Сокращение, условное обозначение	Полное наименование
ТП ПИТ	Требование прерывания по нарушению питания
ТПР	Требование прерывания
ТП РГН	Требование прерывания по регенерации
ТП ТМ	Требование прерывания по таймеру
ТС	Тактирующий сигнал
ТТЛ	Транзистор - транзисторная логика
УВВИ	Устройство ввода-вывода информации
УС	Указатель стека
УСТ РГН	Установка триггера РЕГЕНЕРАЦИЯ
Ф	Фаза (тактовый сигнал)
ЦП	Центральный процессор
ЦПД	Цикл прямого доступа к памяти
ЭВМ	Электронно-вычислительная машина
ЭПМ	Электрифицированная пишущая машина

2. НАЗНАЧЕНИЕ

2.I. Центральный процессор предназначен для обработки информации и может быть использован в составе ЭВМ:

- в системах управления технологическими процессами,
- в составе испытательного и контрольно-измерительного оборудования,
- для научно-технических и экономических расчетов.

2.2. ЦП рассчитан на эксплуатацию в климатических условиях группы 2 по ГОСТ I6325-76.

2.3. Питание ЦП осуществляется от источников постоянного тока напряжением плюс 5 В ± 0,25 В и плюс I2 В ± 0,36 В.

3. ТЕХНИЧЕСКИЕ ДАННЫЕ

3.I. Габаритные размеры и масса ЦП соответствуют указанным в табл. 2.

Таблица 2

Наименование	Габаритные размеры, мм	Масса, кг не более
Центральный процессор М2	252 х 296 х I2	0,8

3.2. Максимальный электрический ток, потребляемый ЦП:

от источника плюс 5 В не более 2,5 А,

от источника плюс I2 В не более I,5 А.

3.3. Основные характеристики центрального процессора

3.3.I. Система счисления для чисел и команд - двоичная.

3.3.2. Разрядность для чисел и команд - I6 двоичных разрядов.

3.3.3. Принцип работы основных устройств - параллельный.

3.3.4. Система команд - безадресная, одноадресная, двухадресная.

3.3.5. Виды адресации: регистровая, косвенно-регистровая,

автоинкрементная, косвенно-автоинкрементная, автодекрементная, косвенно-автодекрементная, индексная, косвенно-индексная.

3.3.6. Число регистров общего назначения - 8.

3.3.7. Количество каналов передачи информации - I.

3.3.8. Количество уровней запроса канала внешними устройствами - 3.

3.3.9. Количество уровней запроса канала, на которых возможно прерывание программы - 2, в том числе один уровень запроса от таймера с постоянным адресом вектора прерывания.

3.3.I0. Обработка внешних и внутренних прерываний выполняется с помощью памяти магазинного типа (стека).

3.3.II. Количество команд - 72

3.3.I2. Количество двухадресных команд - 8.

3.3.I3. Количество команд расширенной арифметики (РА) - 4.

3.3.I4. Количество команд для вычислений с плавающей запятой (ПЗ) - 4.

3.3.I5. Емкость резидентного оперативного запоминающего устройства (ОЗУ) - 4К I6-разрядных слов.

3.3.I6. Время обращения к ОЗУ - не более 700 нс.

3.3.I7. Длительность цикла работы ОЗУ - не более 2,4 мкс.

3.3.I8. Время выполнения двухадресных команд типа "Сложение" при регистровом методе адресации - 4 мкс.

3.3.I9. Время выполнения одноадресных команд типа "Очистка" при регистровом методе адресации - 4,4 мкс.

3.3.20. Время выполнения одноадресных команд типа "Очистка" при косвенно-регистровом методе адресации - 7,6 мкс, при длительности тактирующего импульса 400 нс.

3.3.2I. Время выполнения двухадресных команд типа "Сложение" с плавающей запятой - не более 2I8 мкс.

3.3.22. Время выполнения двухадресных команд типа "Умножение" с плавающей запятой не более - 360 мкс.

3.4. ЦП нормально работает при изменении питающих напряжений в следующих пределах:

по источнику плюс 5 В на ± 0,25 В

по источнику плюс I2 В на ± 0,36 В

3.5. ЦП нормально функционирует при климатических условиях согласно ГОСТ I6325-76, указанных в табл. 3.

Таблица 3

Наименование	Температура окружающего воздуха, °С	Относительная влажность окружающего воздуха при температуре + 30 °С, %	Атмосферное давление, кПа (мм рт.ст.)
Центральный процессор	от плюс 5 до плюс 40	от 40 до 90	от 84,0 до I06,7 (от 630 до 800)

4. УСТРОЙСТВО И РАБОТА ЦП

4.I. Общая теория работы

ЦП является основным устройством в ЭВМ, для его успешного изучения необходимо ознакомиться с общей теорией работы ЭВМ.

ЭВМ имеет модульный принцип построения, т.е. все функциональные блоки ЭВМ выполнены в виде конструктивно законченных устройств (модулей), связь между которыми осуществляется через единый канал обмена информацией. Таким образом, ЭВМ представляет собой систему модулей, объединенных каналом. Пользователю предоставляется возможность самому определить необходимую конфигурацию системы в зависимости от конкретного применения ЭВМ.

Поскольку связь между отдельными элементами системы, включая центральный процессор, осуществляется через канал одинаково, внешние устройства так же легко доступны для ЦП, как и оперативная память.

При построении системы на базе ЭВМ пользователь может подключить к каналу и свои собственные устройства, разработанные с учетом требований и рекомендаций, которые приводится в настоящем ТО

4.I.I. Канал обмена информацией

Канал обмена информацией (канал ЭВМ) является простой, быстродействующей системой связей, соединяющей центральный процессор, память и все внешние устройства.

На рис. I представлена упрощенная блок-схема ЭВМ

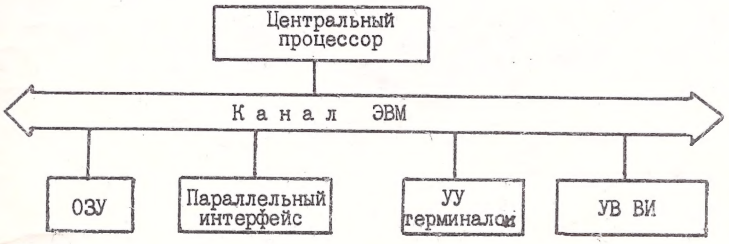


Рис. I Блок-схема ЭВМ

Все модули, подключенные к каналу ЭВМ, используют одни и те же канальные связи.

Канал ЭВМ содержит 38 линий связи, из которых 3I линия являются двунаправленными. Это означает, что по одним и тем же линиям информация может как приниматься, так и передаваться относительно одного и того же устройства.

4.I.I.I. Связь типа "управляющий-управляемый".

Связь между двумя устройствами, подключенными к каналу, осуществляется по принципу "управляющий-управляемый" (активный-пассивный). В любой момент времени только одно устройство является активным. Активное устройство управляет циклами обращения к каналу, удовлетворяет, если это необходимо, требования прерывания от внешних устройств, контролирует предоставление прямого доступа к памяти. Пассивное устройство (управляемое) является только исполнительным устройством. Оно может принимать или передавать информацию только под управлением активного устройства. Типичным примером таких отношений является центральный процессор, как активное устройство, выбирающий команду из памяти, которая всегда является пассивным устройством. Другим примером может служить устройство, работающее в режиме прямого доступа к памяти (устройство ПДП) как активное устройство и память как пассивное устройство (например, контроллер НМД-ОЗУ). В этом случае процессор передает управление каналом устройству ПДП.

4.I.I.2. Замкнутая (асинхронная) связь.

Связь через канал замкнута, т.е. на управляющий сигнал передаваемый активным устройством, должен поступить ответный

сигнал от пассивного устройства. Поэтому процесс обмена между устройствами не зависит от длины канала и времени отклика пассивного устройства (в пределах 10 мкс.).

Асинхронное выполнение операции передачи данных устраняет необходимость в тактовых импульсах. В результате этого обмен с каждым устройством может происходить с максимально возможным для данного устройства быстродействием

Обмен между двумя устройствами может выполняться как 16-разрядными словами, так и байтами (8 разрядов).

4.I.I.3. Обмен данными между устройством пользователя и ЭВМ.

Канал ЭВМ обеспечивает три типа обмена данными – это программный обмен, обмен в режиме ПДП и обмен в режиме прерывания программы.

Программный обмен данными – это передача данных по инициативе и под управлением программы. Обычно, перед началом обмена проверяется содержимое регистра состояния устройства, чтобы определить, готово ли оно к обмену данными. Обмен данными по инициативе внешнего устройства может выполняться при прямом доступе к памяти (режим ПДП) и в режиме прерывания программы.

Обмен в режиме ПДП является самым быстрым способом передачи данных между памятью и внешним устройством. Он не меняет состояние ЦП и поэтому может выполняться в промежутках между циклами обращения к каналу, проводимых в ЦП.

При этом адресация и управление размерами передаваемого массива данных находятся под управлением устройства, получив-

шего прямой доступ к памяти. Массивы данных в режиме ПДП могут передаваться со скоростью, определяемой быстродействием памяти.

Обмен данными в режиме прерывания программы – это выполнение программы обслуживания по требованию периферийного устройства. ЦП при этом приостанавливает выполнение текущей программы, чтобы обслужить запрашивающее устройство. После завершения выполнения программы обслуживания ЦП возобновляет выполнение прерванной программы с того места, где она была прервана.

4.I.I.4. Сигналы канала

В таблице 4 приведены наименования сигналов, их обозначения и соответствующие им контакты коммутационной платы субблока.

Конструктивно канал ЭВМ выполнен в виде печатной платы, обеспечивающей необходимые электрические соединения между контактами розеток, к которым подключаются различные устройства. Расположение устройств на коммутационной панели может быть произвольным, поэтому в таблице 4 не указываются номера гнезд (+AI:AI, +A3:AI и т.д.). Кроме того, гнезда В и Г идентичны гнездам А и Б, поэтому в таблицу включены только контакты гнезд А и Б.

Обозначение контактов коммутационной панели и печатной платы приведены на рис. 2 и 3.

Таблица 4

Номер контакта	Обозначение сигнала	Наименование сигнала
+А:А1	К РД1	Резервные линии канала. (Зарезервированы
+А:А2	К РД2	для дальнейших расширений системы. Поль-
+А:А3	К РД3	зователю не рекомендуется использовать в
+А:А4	К РД4	своих целях).
+А:А5	РЕЗ1	Резервные контакты, с каналом не соеди-
+А:А6	РЕЗ2	няются (Могут быть задействованы поль-
+А:А7	РЕЗ3	зователем).
+А:А8	ОБЩ	Общий
+А:А9	ОТЛ1	Технологическая линия отладки, на ком-
+А:А10	ОТЛ1	мутационной панели К1 эти контакты объ- единены между собой
+А:А11	ОБЩ	Общий
+А:А12	К ПДД Н	Требование прямого доступа к памяти
+А:А13	К ОСТ Н	Останов
+А:А14	К РГН Н	Регенерация
+А:А15	ПРЕЗ 1	Резервный контакт. (Использовать не ре- комендуется).
+А:А16	ОБЩ	Общий
+А:А17	ПРЕЗ 2	Резервный контакт. (Использовать не ре- комендуется).
+А:А18	+5ВБ	Батарейное питание +5 В.
+Б:А1	К ПОСТН В	Постоянное питание нормально
+Б:А2	К ПИТН В	Сетевое питание нормально
+Б:А3	РЕЗ4	Резервные контакты, с каналом не соеди-

Продолжение табл. 4

Номер контакта	Обозначение сигнала	Наименование сигнала
		няются.
+Б:А4	РЕЗ5	(Могут быть задействованы пользователем)
+Б:А5	РЕЗ6	
+Б:А6	РЕЗ7	
+Б:А7	РЕЗ8	
+Б:А8	ОБЩ	Общий
+Б:А9	ОТЛ 2	Технологическая линия отладки, на
+Б:АI0	ОТЛ 2	коммутационной панели КП эти контакты объединены между собой.
+Б:АII	ОБЩ	Общий
+Б:АI2	К ПВ Н	Подтверждение выбора
+Б:АI3	К РЛ 5	Резервная линия канала (Зарезервирована для дальнейших расширений системы. Пользователю не рекомендуется использовать в своих целях).
+Б:АI4	К ПРТ Н	Требование прерывания по внешнему событию (таймеру).
+Б:АI5	ПРЕЗ 3	Резервный контакт. (Использовать не рекомендуется).
+Б:АI6	ОБЩ	Общий
+Б:АI7	ПРЕЗ 4	Резервный контакт. (Использовать не рекомендуется).
+Б:АI8	+5В	Питание +5В.
+А:БI	+5В	Питание +5 В

Продолжение табл. 4

Номер контакта	Обозначение сигнала	Наименование сигнала
+А:Б2	-I2 В	Этот контакт зарезервирован под питание -I2В, если в системе используется этот номинал напряжения питания. Однако, он может быть использован и для других цепей питания пользователя, если это необходимо
+А:Б3	ОБЩ	Общий
+А:Б4	+I2 В	Питание +I2 В
+А:Б5	К ВЫВОД Н	Вывод данных
+А:Б6	К СИП Н	Сигнал синхронизации пассивного устройства
+А:Б7	К ВВОД Н	Ввод данных
+А:Б8	К СИА Н	Сигнал синхронизации активного устройства
+А:Б9	К БАЙТ Н	Вывод байта
+А:БI0	К ТПР Н	Требование прерывания
+А:БI1	К ППР I Н	Входной и выходной сигналы предоставления прерывания
+А:БI2	К ППР О Н	
+А:БI3	К ВУ Н	Выбор внешнего устройства
+А:БI4	К ППД I Н	Входной и выходной сигналы предоставления прямого доступа к памяти
+А:БI5	К ППД Г Н	
+А:БI6	К СБРОС Н	Первоначальная установка канала

Номер контакта	Обозначение сигнала	Наименование сигнала
+А:БI7	К ДА 00 Н	
+А:БI8	К ДА 0I Н	Линии адреса (данных)
+Б:БI	+5 В	Питание +5 В
+Б:Б2	-I2В	Питание -I2В (если необходимо)
+Б:Б3	ОБЩ	Общий
+Б:Б4	+I2В	Питание +I2 В
+Б:Б5	К ДА 02 Н	
+Б:Б6	К ДА 03 Н	
+Б:Б7	К ДА 04 Н	
+Б:Б8	К ДА 05 Н	
+Б:Б9	К ДА 06 Н	
+Б:БI0	К ДА 07 Н	
+Б:БI1	К ДА 08 Н	
+Б:БI2	К ДА 09 Н	Линии адреса (данных)
+Б:БI3	К ДА I0 Н	
+Б:БI4	К ДА II Н	
+Б:БI5	К ДА I2 Н	
+Б:БI6	К ДА I3 Н	
+Б:БI7	К ДА I4 Н	
+Б:БI8	К ДА I5 Н	

Как адрес, так и данные (слова или байты) передаются по одним и тем же I6 линиям адреса/данных К ДА (00-I5) Н. Любой цикл обращения к каналу начинается с адресации пассивного устройства. После завершения адресной части цикла ((400нс),

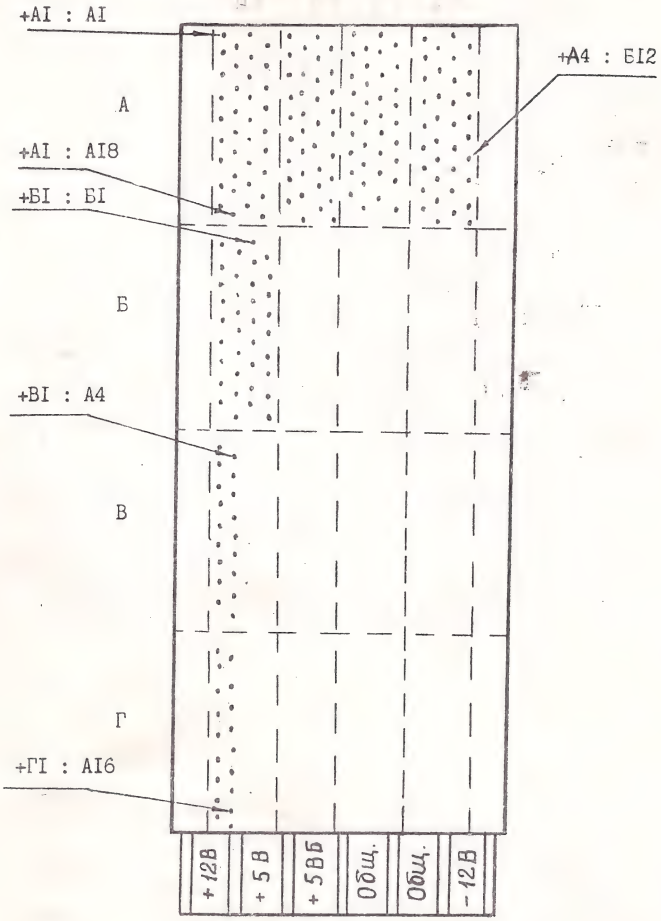


Рис. 2.Обозначение контактов коммутационной панели

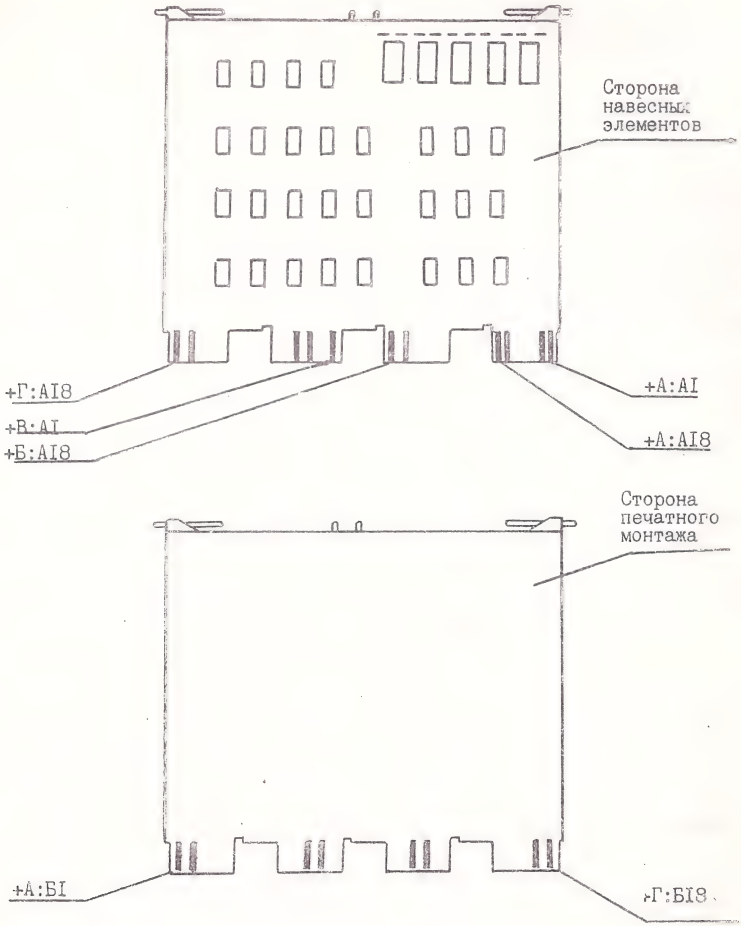


Рис.3. Обозначение контактов печатной платы.

активное устройство выполняет прием или передачу данных, которые выполняются асинхронно и требуют ответа от адресуемого устройства.

Функции синхронизации при передаче адреса и приеме/передачи данных выполняют сигналы управления каналом К СИА Н, К БАЙТ Н, К ВВОД Н, К ВЫВОД Н и К СИП Н.

Сигнал синхронизации активного устройства (К СИА Н) вырабатывается активным устройством. Передний фронт сигнала означает, что адрес находится на линиях К ДА (00÷15) Н. Сигнал К СИА Н сохраняет активный уровень до окончания текущего цикла обращения к каналу.

Сигнал синхронизации пассивного устройства (К СИП Н) информирует активное устройство о том, что данные приняты с линий К ДА (00÷15) Н или данные установлены на информационных линиях. Этот сигнал вырабатывается в ответ на сигналы К ВВОД Н и К ВЫВОД Н.

Сигнал К ВВОД Н используется в двух случаях:

а) если он вырабатывается во время действия сигнала К СИА Н, он означает ввод данных по отношению к активному устройству. Он вырабатывается, когда активное устройство готово принять данные от пассивного;

б) если он вырабатывается с сигналом К ППР Н (сигнал К СИА Н – пассивный), это означает, что выполняется операция ввода адреса вектора при прерывании программы.

Сигнал КВЫВОД Н означает, что по отношению к активному устройству выполняется операция вывода, и на линиях К ДА

(00÷15) Н помещены истинные данные. Выработка сигнала К ВЫВОД Н активным устройством задерживается по отношению к подаваемым в канал данным не менее, чем на 150 нс. Пассивное устройство, отвечая на сигнал К ВЫВОД Н, должно вырабатывать ответный сигнал К СИП Н, чтобы завершить операцию передачи данных.

Сигнал вывода байта (К БАЙТ Н) используется в двух случаях:

а) он вырабатывается в адресной части цикла для указания, что далее следует операция вывода (ВЫВОД или ВЫВОД Б), а не ввода,

б) он вырабатывается при передаче данных в цикле ВЫВОД Б для указания, что выводится байт.

Сигнал требования прерывания (К ТПР Н) вырабатывается пассивным устройством, если его триггеры требования прерывания и разрешения прерывания установлены. Этот сигнал информирует процессор о том, что устройство готово передавать или принять данные. Если 7 разряд ССП установлен в 0, процессор разрешит прерывание, вырабатывая сигналы К ВВОД и К ППР Н.

Входной сигнал предоставления прерывания (К ППР I Н) и выходной сигнал предоставления прерывания (К ППР 0 Н) вырабатываются следующим образом.

В ответ на требование прерывания (К ТПР Н) ЦП вырабатывает сигнал К ППР Н, который является выходным для ЦП и входным для первого устройства в канале (электрически ближе расположенного к ЦП, следовательно, имеющего более высокий приоритет). Если это устройство не требовало прерывания (не вырабатывает сигнал К ТПР Н), то оно пропустит сигнал К ППР Н к следующему устройству. При этом сигнал К ППР Н будет выход-

ным по отношению к первому устройству и входным по отношению ко второму и так далее. (Схему распространения сигнала К ПРР Н см. рис. 4). Устройство, которое требовало прерывания, запретит распространение этого сигнала. Таким образом, линия сигнала канала К ПРР Н, последовательно проходя через все устройства, обеспечивает их поочередный опрос и, следовательно, различный приоритет обслуживания.

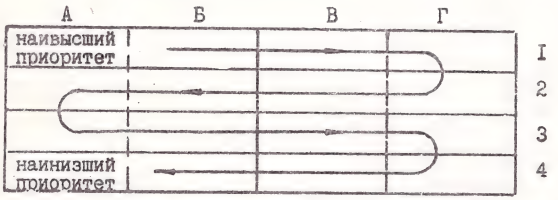


Рис. 4. Схема распространения сигналов К ПРР Н и К ПЦД Н на монтажной панели субблока

Процедура предоставления прямого доступа к памяти (ПДП) выполняется под управлением трех сигналов: К ТПД Н, К ПЦД Н и К ПВ Н. Устройство вырабатывает сигнал требования прямого доступа к памяти (К ТПД Н), требуя передачи ему канала, и ожидает окончания текущего цикла обращения к каналу. После снятия сигнала К СИА Н, ЦП вырабатывает сигнал К ППД Н. Сигнал предоставления прямого доступа к памяти (К ППД Н) последовательно проходит через внешние устройства и предоставляет канал устройству с наивысшим приоритетом, запросившему прямой доступ к памяти. Если первое устройство запрашивало канал, он запретит передачу сигнала К ПЦД Н, если нет – передает

этот сигнал к следующему устройству и т.д. Устройство, запросившее канал, отвечает на сигнал К ППД Н выработкой сигнала К ПВ Н и снятием сигнала К ТПД Н.

Сигнал К ПВ Н вырабатывается устройством ПДП в ответ на процессорный сигнал К ППД Н и говорит о том, что устройство приняло управление каналом на себя. После этого устройство ПДП может производить обмен данными, используя для этого стандартные циклы обращения к каналу.

Следует заметить, что устройство ПДП не должно мешать регенерации ОЗУ (должно обмениваться одиночными словами), или оно должно брать функцию регенерации на себя.

Сигнал К ПРТ Н – требование прерывания по внешнему событию (таймеру). Когда он вырабатывается, процессор переходит (если 7 разряд ССП устновлен в 0) на программу обслуживания через вектор с адресом 100_8. Обычно этот сигнал вырабатывается таймером, работающим от импульсов сетевого питания. Требование прерывания от таймера имеет более высокий приоритет по сравнению с обычным требованием прерывания от внешнего устройства.

Следует заметить, что пользователь может использовать линию К ПРТ Н не только для прерывания от таймера. Если это необходимо, на эту линию может быть подключено любое устройство. Сигнал К ПРТ Н можно запрещать с помощью перемычки на модуле ЦП.

Сигнал выбора внешнего устройства (К ВУ Н) вырабатывается активным устройством, когда в канал передается адрес, относящийся к последним 4К адресов (с 160000 по 177777). Сигнал

К ВУ Н остается активным на время адресной части цикла обращения к каналу.

Сигнал К РТИ Н вырабатывается при регенерации памяти. Этот сигнал заставляет все блоки динамической памяти выполнять построчную регенерацию при каждой канальной операции К СИА Н/К ВВОД Н. Процессору можно запретить управление регенерацией путём установки перемычки на модуле ЦП. Тогда регенерацией может управлять внешнее устройство или она будет запрещена, если в системе не используется динамическая память.

Сигнал К ПОСТН В (постоянное питание нормальное) вырабатывается источником питания, если постоянные напряжения питания достаточны для обеспечения надежной работы системы.

Сигнал К ПИТН В (питание нормальное) вырабатывается источником питания, если сетевое напряжение питания нормально. Если этот сигнал становится пассивным во время работы процессора, то он вызывает прерывание программы по нарушению питания.

Сигнал К СБРОС Н вырабатывается процессором, чтобы выполнить начальную установку (СБРОС) всех устройств, подключенных к каналу. Этот сигнал вырабатывается при каждом включении питания по сигналу К ПОСТН В. Сигнал К СБРОС Н вырабатывается программно по команде РЕБЕТ и при пуске программы (нажатие клавиши "G").

При подаче команды "G" возможно появление символа "Забой", не влияющего на общую работоспособность ЭВМ.

Канальным сигналом К ССТН (останов) процессор переводится в режим связи с пультовым терминалом. При этом ЦП прекращает выполнение текущей программы и игнорирует все внешние запросы прерывания. В этом режиме процессор микропрограммно

выполняет команды, подаваемые с любого пультового терминала, имеющего адрес 177560.

4.1.1.5. Циклы обращения к каналу

Для выполнения любой команды процессору требуется выполнить хотя бы одну операцию обращения к каналу. Для некоторых команд требуется выполнить несколько операций. Первой такой операцией для всех команд является ввод данных из ячейки памяти, адрес которой определяется счетчиком команд (СК). Все операции обращения к каналу для ввода и вывода данных называются циклами обращения к каналу. Если для выполнения команды не требуется обращаться за операндами к памяти или к внешним устройствам, дополнительных циклов канала не требуется. Однако, если выполняется команда с обращением к памяти или устройствам, то в этом случае могут выполняться любые из следующих циклов: 1) ВВОД, 2) ВВОД-ПАУЗА-ВЫВОД, 3) ВЫВОД, 4) ВЫВОД Б.

В промежутках между циклами обращения к каналу ЦП может предоставлять канал устройству ПДП. Требование прерывания может быть удовлетворено только перед выборкой команды (т.е. в промежутках между выполнением команд).

Ниже описываются различные типы канальных операций. Следует заметить, что последовательность операций при выполнении обмена данными между ЦП и памятью аналогичны последовательности при выполнении обмена между ЦП и внешним устройством. Цикл ВЫВОД (или ВЫВОД Б) аналогичен операции записи, а цикл ВВОД – считыванию. Кроме того, цикл ВВОД-ПАУЗА-ВЫВОД включает ввод данных, выполнение арифметическо-логических операций

и вывод результата операций без повторений передачи адреса, то есть результат записывается по адресу последнего выбранного операнда.

Цикл ВВОД

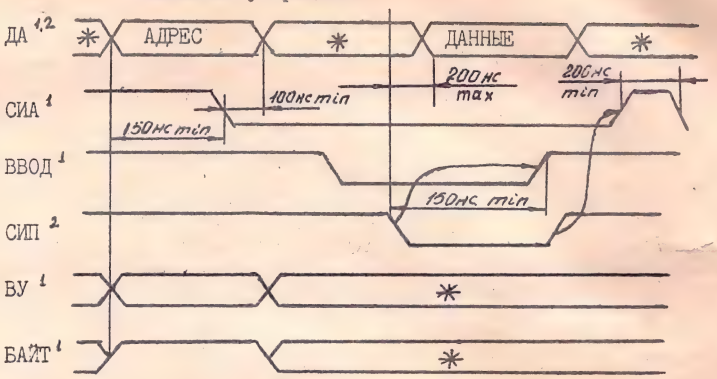
Направление передачи при выполнении операций обмена данными определяется по отношению к активному устройству. При выполнении цикла ВВОД данные передаются от пассивного устройства к активному.

Временная диаграмма выполнения цикла ВВОД представлена на рис. 5.

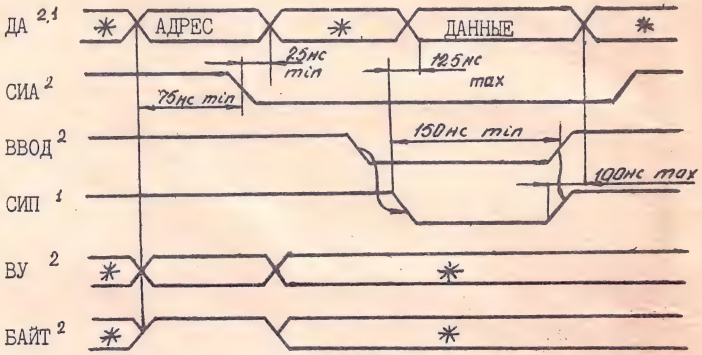
Порядок выполнения операций следующий:

- активное устройство в адресной части передает по линиям К ДА (00÷15) Н адрес, а также вырабатывает сигнал К ВУ Н, если адрес находится в диапазоне 160000 ÷ 177777.

Активное устройство



Пассивное устройство



I - передаваемый сигнал
2 - принимаемый сигнал
✱ - уровень сигнала не имеет значения

Рис. 5. Временная диаграмма цикла ВВОД

Не менее чем через 150 нс после установки адреса активное устройство вырабатывает сигнал К СИА Н, предназначенный для запоминания адреса во входной логике выбранного устройства;

- пассивное устройство дешифрирует адрес и запоминает его,

- активное устройство снимает адрес с линий К ДА (00-15)Н, очищает линию К ВУ Н и вырабатывает сигнал К ВВОД Н, сигнализируя о том, что оно готово принять данные от пассивного устройства и ожидает поступления сигнала К СИП Н,

- пассивное устройство помещает данные на линии К ДА (00-15)Н и вырабатывает сигнал К СИП Н, сигнализирующий о том, что данные находятся в канале. Если сигнал К СИП Н не вырабатывается в течение 10 мкс после выработки сигнала К ВВОД Н, ЦП переходит к обслуживанию внутреннего прерывания по ошибке обращения к каналу с адресом вектора 4,

- активное устройство принимает сигнал К СИП Н, принимает данные, снимает сигнал К ВВОД Н,

-пассивное устройство снимает сигнал К СИП Н, завершая операцию передачи данных,

- активное устройство снимает сигнал К СИА Н по заднему фронту сигнала К СИП Н, завершая тем самым канальный цикл ВВОД.

Следует отметить, что во время выполнения цикла ВВОД сигнал К БАЙТ Н не вырабатывается.

Цикл ВЫВОД

При выполнении цикла ВЫВОД данные передаются от активного устройства к пассивному, например, происходит запись данных в память.

Временная диаграмма выполнения цикла ВЫВОД представлена на рис. 6.

Порядок выполнения операций следующий:

– активное устройство в адресной части цикла передает по линиям К ДА (00–15) Н адрес, а также сигнал К ВУ Н, если это необходимо.

Кроме того, в цикле ВЫВОД в адресной части всегда вырабатывается сигнал К БАЙТ Н. Не менее чем через 150 нс (минимум) после установки адреса вырабатывается сигнал К СИА Н. Функции, выполняемые этими двумя сигналами, те же, что и в цикле ВВОД:

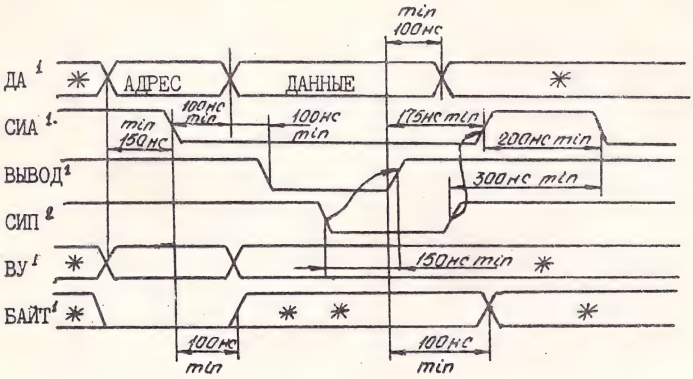
– пассивное устройство дешифрирует адрес запоминает его;

– активное устройство снимает адрес с линий К ДА (00–15)Н, очищает линию К ВУ Н и снимает сигнал К БАЙТ Н. После этого активное устройство помещает данные на линии К ДА (00–15) Н и через 100 нс (минимум) вырабатывает сигнал К ВЫВОД Н, означающий, что на линиях К ДА (00–15)Н помещены данные;

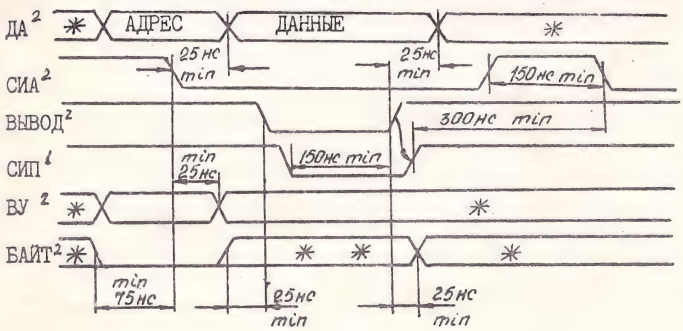
–пассивное устройство принимает данные с линий К ДА (00–15)Н и вырабатывает сигнал К СИП Н, означающий, что данные приняты пассивным устройством. Если сигнал К СИП Н не вырабатывается в течение 10 мкс после выработки сигнала К ВЫВОД Н, процессор переходит к обслуживанию внутреннего прерывания по ошибке обращения к каналу с адресом вектора 4;

– активное устройство, получив сигнал К СИП Н, очищает через 150 нс (минимум) линию К ВЫВОД Н, а через 250 нс (минимум) после поступления сигнала К СИП Н с линий К ДА (00–15)Н снимаются данные. Подобная последовательность операций обеспечивает надежный прием данных пассивным устройством;

Активное устройство



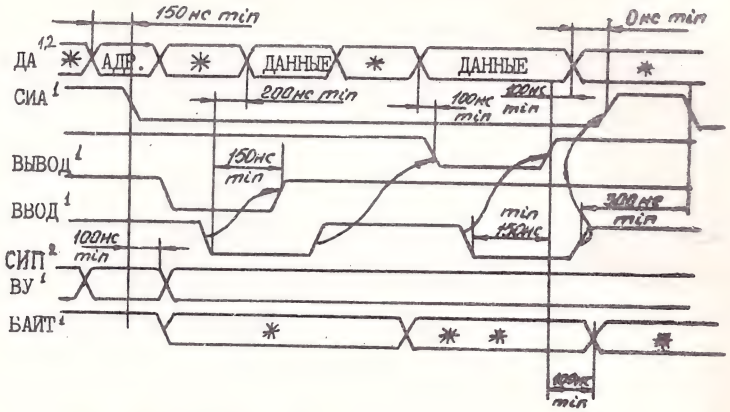
Пассивное устройство



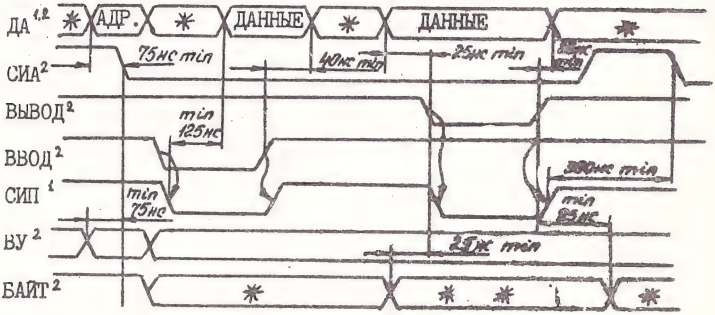
I - передаваемый сигнал
2 - принимаемый сигнал
* - уровень сигнала не имеет значения
* * - устанавливается при байтовых операциях

Рис. 6. Временная диаграмма цикла ВЫВОД.

Активное устройство



Пассивное устройство



1 - передаваемый сигнал
2 - принимаемый сигнал
✳ - уровень сигнала не имеет значения
✳✳ - устанавливается при байтовых операциях

Рис. 7 Временная диаграмма цикла ВВОД-ПАУЗА-ВЫВОД

– пассивное устройство снимает сигнал К СИП Н, завершая операцию приема данных;

– активное устройство снимает сигнал К СИА Н, завершая цикл канала ВЫВОД.

Следует заметить, что сигнал К БАЙТ Н в части передачи данных может быть как пассивным, так и активным, определяя тем самым вывод 16-разрядного слова или вывод байта.

ЦИКЛ ВВОД-ПАУЗА-ВЫВОД

Цикл обращения к каналу ВВОД-ПАУЗА-ВЫВОД аналогичен операции СЧИТЫВАНИЕ-МОДИФИКАЦИЯ-ЗАПИСЬ.

Временная диаграмма выполнения цикла ВВОД-ПАУЗА-ВЫВОД представлена на рис. 7.

Адресная часть и ввод данных выполняются аналогично циклу ВВОД. Однако, сигнал К СИА Н остается активным и после окончания ввода данных, что позволяет осуществлять вывод и модификацию данных без повторений адресной части цикла.

Следует заметить, что операция по выводу данных может быть байтовой, поэтому сигнал К БАЙТ Н в это время может быть как пассивным, так и активным.

4.I.I.6. Прямой доступ к памяти

В режиме прямого доступа к памяти обмен данных происходит между периферийным устройством и памятью без вмешательства ЦП. Устройство, способное работать в режиме ПДП (устройство ПДП), должно выполнять все функции активного устройства: адресацию, синхронизацию, выработку управляющих сигналов и, если это необходимо, управление регенерацией ОЗУ.

Обмен данными в режиме ПДП производится стандартными циклами обращения к каналу (п. 4.I.I.5).

Временная диаграмма предоставления прямого доступа показана на рис. 8.

Порядок выполнения операций следующий:

– устройство запрашивает канал, вырабатывая сигнал требования прямого доступа к памяти (К ТПД Н),

– после завершения текущего цикла канала ЦП вырабатывается сигнал предоставления прямого доступа к памяти (К ППД Н), запрещая выработку следующего "процессорного" цикла канала:

– устройство ПДП получает сигнал К ППД Н, вырабатывает сигнал подтверждения выбора (К ПВ Н) и снимает сигнал К ТПД Н,

– ЦП снимает сигнал К ППД Н и ожидает завершения операций ПДП,

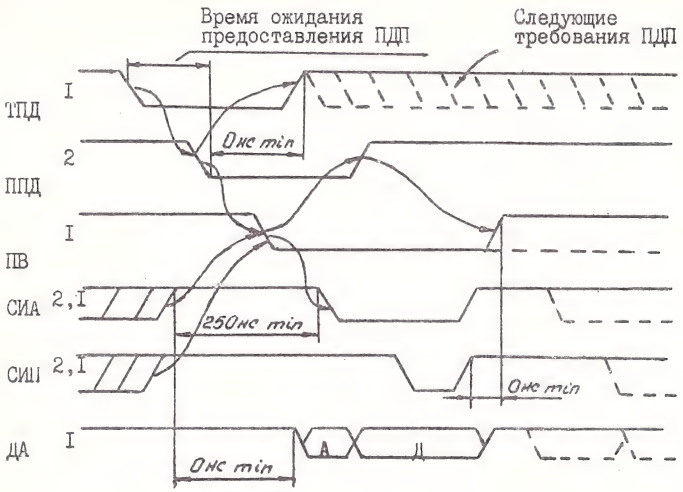
– устройство ПДП получает канал и выполняет требуемые циклы передачи данных таким же образом, как это описано для операции ВВОД, ВВОД-ПАУЗА-ВЫВОД и ВЫВОД.

Когда передача данных закончится, устройство снимает сигнал К ПВ Н и возвращает управление каналом процессору,

– ЦП возобновляет свою работу, вырабатывая сигнал К СИА Н или выдает новое разрешение использования канала, если вырабатывается сигнал К ТПД Н.

4.I.I.7. Прерывание программы

Прерывание программы – это временное прекращение выполнения текущей программы и переход к выполнению программы обслуживания устройства, вызвавшего прерывание. Каждое устройство, способное вызывать прерывание, должно иметь регистр состояния, содержащий разряд разрешения прерывания. Этот разряд должен быть программно установлен в "I", если устройству разрешается прерывание программы. Кроме того, ЦП будет



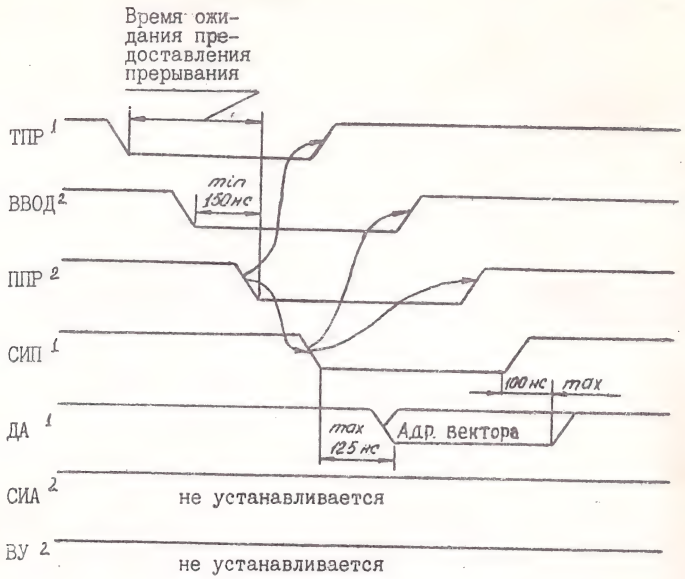
I – передаваемый сигнал устройством ПДП

2 – принимаемый сигнал устройством ПДП

А – адрес

Д – данные

Рис. 8 Временная диаграмма предоставления прямого доступа к памяти.



I - сигнал, передаваемый устройством

2 - сигнал, принимаемый устройством

Рис. 9. Временная диаграмма прерывания программы.

предоставлять прерывание, и, следовательно, выполнять программу обслуживания устройства только в том случае, если его 7 разряд в РСП будет содержать 0. Каждое устройство, требующее прерывания, имеет программу обслуживания, вход в которую осуществляется автоматически с помощью вектора прерывания.

Временная диаграмма операций по прерыванию программы представлена на рис. 9.

Последовательность операций следующая:

– устройство, которому необходимо обслуживание, вырабатывает сигнал К ТПР Н,

– процессор удовлетворяет требование, вырабатывая сигналы К ВВОД Н и К ШПР Н,

–устройство получает сигнал К ВВОД Н и входной сигнал предоставления прерывания (К ШПР I Н) и запрещает распространение этого сигнала к другим устройствам. Затем оно помещает адрес вектора прерывания на линии К ДА (00–07) Н, вырабатывает сигнал К СИП Н и снимает сигнал требований прерывания К ТПР Н.

Если сигнал К СИП Н не вырабатывается в течение 10 мкс, ЦП переходит в режим связи с пультовым терминалом:

– процессор принимает адрес вектора прерывания и снимает сигналы К ВВОД Н и К ШПР Н,

– устройство завершает передачу адреса вектора и снимает сигнал К СИП Н,

– ЦП помещает в стек содержимое СК и РСП и загружает новое содержимое в СК и РСП из двух последовательных ячеек, первая из которых определяется адресом вектора прерывания,

после чего переходит к выполнению программы обслуживания данного устройства.

По завершении программы обслуживания ЦП возобновляет выполнение прерванной программы с помощью команды возврата из прерывания (RTI), по которой из стека выбираются два слова и засылаются в СК и РСП.

4.I.I.8. Приоритет устройства

Каждое устройство, способное вызывать прерывание, имеет приоритет обслуживания, основанный на его расположении по отношению к процессору. Когда два (или более) устройства требуют прерывания, устройство, электрически ближе расположенное к ЦП, имеет более высокий приоритет, и оно, получив сигнал предоставления прерывания, запретит его распространение вдоль канала. Любое устройство может вновь прервать выполнение программы обслуживания первого устройства, если при этом 7 разряд ССП содержит 0. Следовательно, программы обслуживания могут "вкладываться" друг в друга до любого уровня.

Поскольку требования прерывания могут одновременно поступить от нескольких устройств, то пользователем устанавливается приоритет (очередность) их обслуживания. Модули интерфейса внешних устройств, а также модули управления вводом-выводом устанавливаются в гнезда монтажной панели в соответствии с их приоритетом.

На рис. I0 показан порядок понижения приоритета от позиции I до позиции 6.

А	Б	В	Г	
Модуль				I
позиция 2		позиция I		2
позиция 3		позиция 4		3
позиция 6		позиция 5		4

Рис. IO. Приоритет устройств согласно их расположению на монтажной панели (вид со стороны установки модулей)

Модуль, установленный в I позицию, имеет наивысший приоритет обслуживания. Факторы, которые следует принимать во внимание при присвоении приоритета внешним устройствам, следующие;

а) быстродействующим устройствам присваивается наивысший приоритет обслуживания,

б) наивысший приоритет обслуживания присваивается также тем устройствам, данные от которых не могут быть восстановлены.

4.I.I.9. Распределение адресов канала

Распределение адресов канала показано на рис. II. Все адреса даны в восьмеричном коде. Буква "К" используется для обозначения числа, равного $1024_{10}(2^{10})$.

Канал ЭВМ позволяет адресоваться к 32К I6-разрядных слов или к 64К байтов.

Ячейки ОЗУ с 000000_8 по 000376_8 зарезервированы под векторы прерывания и использовать их для других целей не рекомендуется. Для каждого вектора необходимы две I6-разрядных ячейки, поэтому адреса векторов прерываний являются четными и

Рис. II. Распределение адресов канала

заканчиваются на 0 или на4.

Последние 4К I6-разрядных адресов окончно отводятся для регистров внешних устройств, поэтому максимальный объем реальной памяти равен 28К I6-разрядных слов.

4.I.I.IO. Уровни сигналов канала

Для всех видов сигналов, передаваемых по линиям канала (за исключением сигналов К ПОСТН В и К ШИН В) уровни "логической I" и "логического 0" следующие:

"логический 0" – плюс 2,0 – 3,4 В

"логическая I" – 0–плюс 0,8 В.

Таким образом, "логический 0" в канале представлен высоким уровнем (пассивное состояние линии), а логическая единица – низким уровнем (активное состояние линии). Для сигналов К ПОСТН В и К ШИН В активным является высокий уровень, а пассивным – низкий уровень.

Чтобы облегчить определение, какой уровень для данного сигнала является активным, в обозначение всех сигналов в настоящем ТО и во всех принципиальных схемах на отдельные устройства входят буквенные обозначения активных уровней. Буква Н в конце наименования сигнала означает активный низкий уровень, буква В – активный высокий уровень. Например, К ВВОД Н и К ПОСТН В.

4.I.I.II. Конструктивное исполнение канала.

Конструктивно канал представляет собой систему проводников в виде печатной платы, с помощью которой соединяются контакты розеток субблока, образуя 38 сигнальных линий и линии питания.

Все линии канала подключаются к согласующим резисторам, которые обеспечивают уровень "логического 0" в канале. Оконечные согласующие делители показаны на рис. I2.

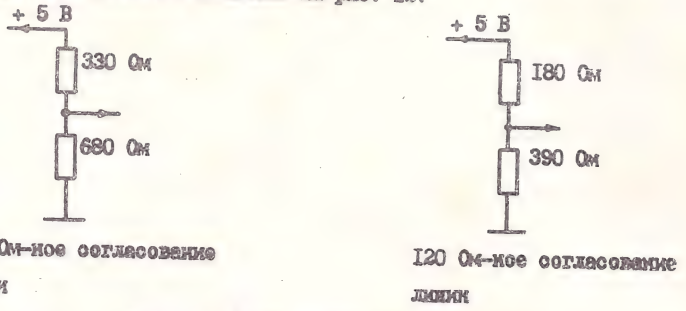


Рис. I2 Схема согласующих делителей

4.I.I.I2. Длина и нагрузочная способность канала

В последующем описании под условной единицей нагрузки понимается подключение одного канального приемника и одного канального передатчика (суммарный ток потребления и ток утечки передатчика не более I05 мкА) и не более I0 пФ емкости к каждой линии канала.

Канал позволяет значительно расширить возможности ЭВМ за счет подключения различных дополнительных устройств. Например, дополнительные блоки ОЗУ, ППЗУ, УВВИ и т.д.

Кроме того, интерфейси нестандартных устройств (устройств пользователя) обычно подключаются непосредственно к каналу, что позволяет сами устройства располагать на любом расстоянии от ЭВМ, так как в этом случае характер связи с устройством будет определять интерфейс, а не ЭВМ.

В связи с тем, что количество устройств, входящих в систему на базе ЦП может быть различным, пользователю рекомендуется руководствоваться одной из трех схем использования канала (в зависимости от количества нагрузок), которые приводятся в этом подпункте. Эти рекомендации позволяют строить системы на базе ЭВМ с использованием одного, двух или трех субблоков.

250 Ом -ное и 120 Ом-ное согласование обеспечивается стандартными канальными коннекторами или аналогичными устройствами пользователя.

Схема согласования канала при построении системы на базе одного субблока показана на рис. 13.

При этом нужно иметь в виду следующее:

а) в процессоре к каналу подключены согласующие резисторы с $Z_н$ = 250 Ом,

б) максимально допустимая длина монтажа при данном согласовании - это длина монтажа одного субблока - 50 см. Каждая линия канала проходит по монтажной панели субблока, включающей четыре ряда гнезд. (А, Б, В, Г) по четыре 36-контактных гнезда в каждом ряду (1, 2, 3, 4),

в) максимальная нагрузка - 6 условных единиц.

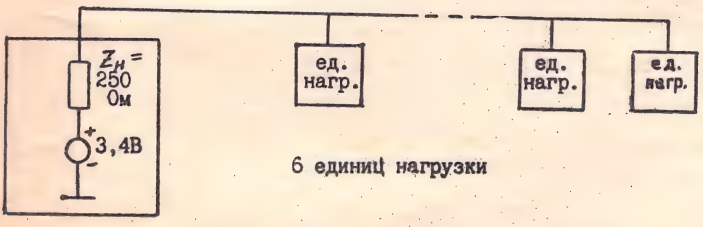


Рис. 13. Схема использования канала с минимальной нагрузкой

Для построения системы на базе двух субблоков следует руководствоваться схемой согласования канала, показанной на рис. I4, и схемой соединения двух субблоков (рис. I5)

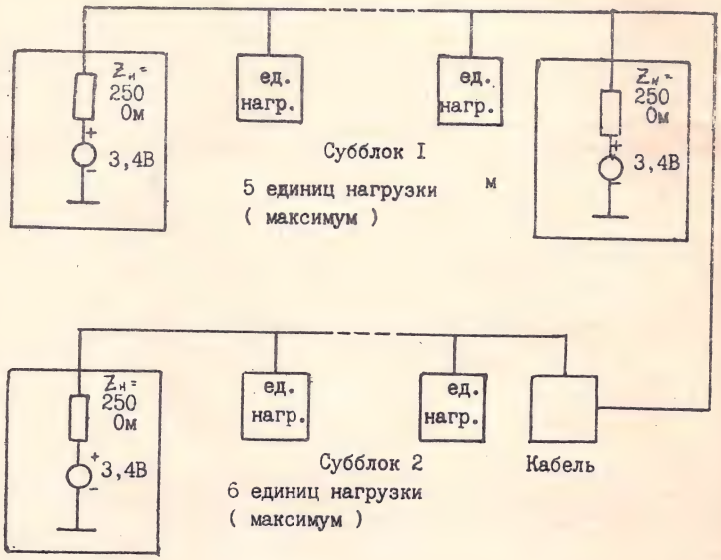


Рис. I4. Схема согласования канала при подключении одного дополнительного субблока

При этом нужно иметь в виду следующее:

а) в процессоре к каналу подключены согласующие резисторы с $Z_н$ = 250 Ом,

б) субблоки соединяются одним кабелем (рис. I5), максимальная длина которого не должна превышать 2,5 м;

в) максимальная нагрузка – II условных единиц;

г) требуется дополнительно согласование с $Z_н$ = I20 Ом (рис. I4).

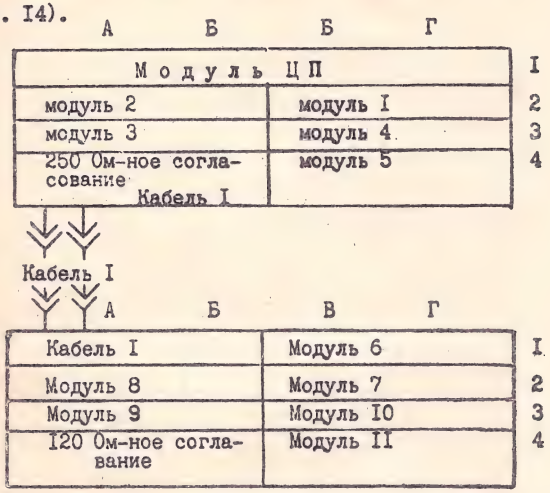


Рис. I5. Схема соединения двух субблоков

Для построения системы с единым каналом на базе трех субблоков следует руководствоваться схемой использования канала с максимальной нагрузкой. При этом нужно иметь в виду следующее:

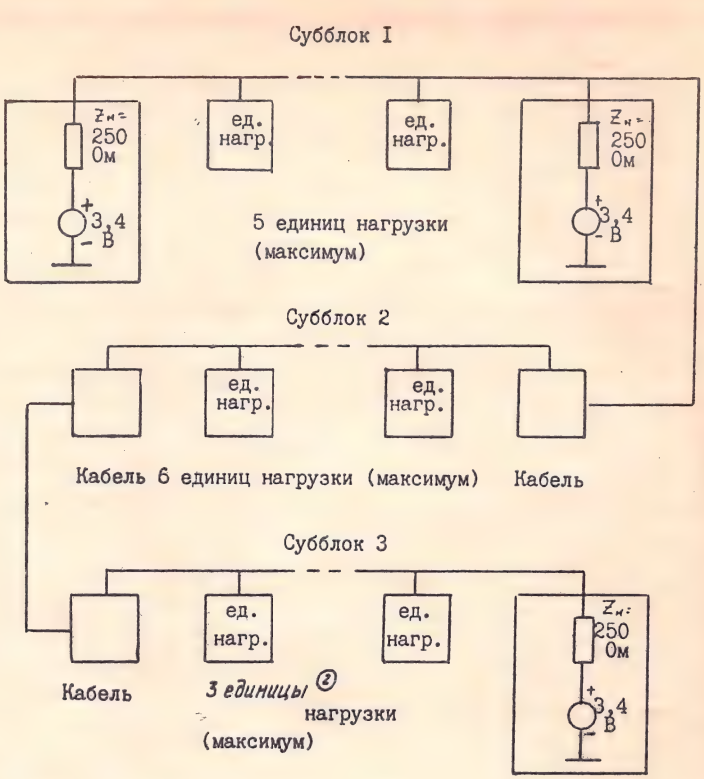


Рис. I6 Схема использования канала с максимальной нагрузкой

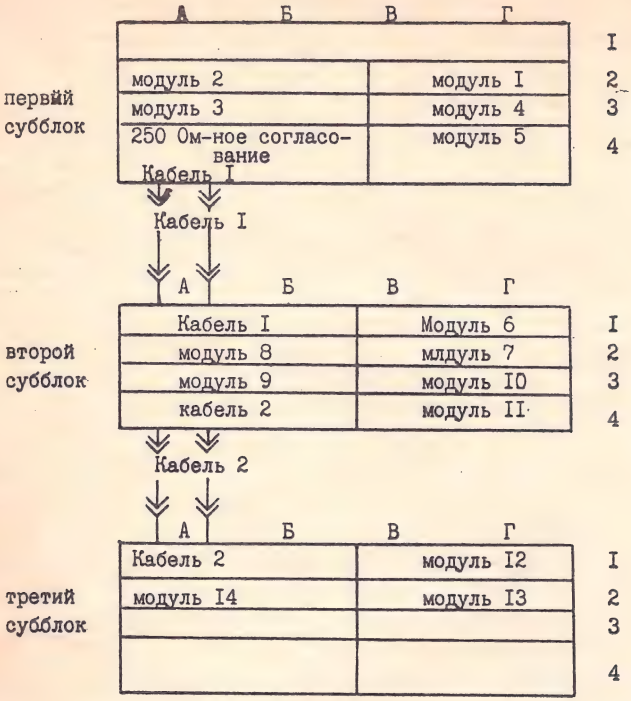


Рис. I7. Схема соединения трех субблоков

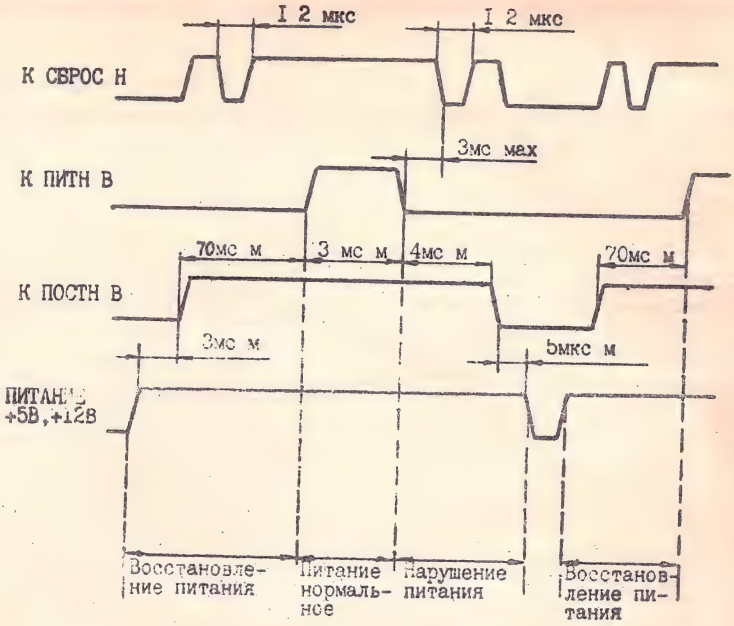


Рис.I8. Временная диаграмма сигналов при нарушении и восстановлении питания.

а) в процессоре к каналу подключены согласующие резисторы с Z_H = 250 Ом,

б) канал образуется объединением трех субблоков двумя кабелями связи. Суммарная длина двух кабелей 3 м.

Максимальная длина монтажа на каждой монтажной панели – 50 см,

в) максимальная нагрузка – 14 условных единиц,

г) требуется согласование линий дополнительно в двух местах (в одном Z_H = 250 Ом, в другом Z_H = 120 Ом, как показано на рис. 16).

При построении систем на базе ЭВМ пользователю следует также руководствоваться рекомендациями, которые приведены в подразделах 4-7 и 4-8.

4.1.1.13. Последовательность операций при восстановлении-нарушении питания

Сигналы К ПИТН В и К ПОСТН В, отражающие состояние питания, должны вырабатываться и сниматься в определенной последовательности по мере того, как подаются или снимаются постоянные напряжения питания.

Временная диаграмма выполнения операций при восстановлении-нарушении питания приведена на рис. 16.

Первоначально сигнал К ПОСТН В и К ПИТН В имеют пассивный (низкий) уровень. Когда постоянные напряжения достигают рабочего уровня, центральный процессор вырабатывает сигнал К СБРОС Н, а приблизительно через 3 мс после этого вырабатывается сигнал К ПОСТ Н В. Центральный процессор отвечает снятием сигнала К СБРОС Н, выполняет микропрограмму выбора режима пуска и ожидает появления сигнала К ПИТН В, который вырабатывается через 70 мс после К ПОСТН В блоком питания

или внешней схемой. После выработки сигнала К ПИТН В заканчивается микропрограмма выбора режима и ЦП переходит к выполнению микропрограммы, соответствующей выбранному режиму.

При нарушении питания блок питания или внешняя схема снимает сигнал К ПИТН В, после чего центральный процессор переходит на подпрограмму обслуживания внутреннего прерывания с адресом вектора 24_8. Через 3 мс (максимум) после снятия сигнала К ПИТН В центральный процессор вырабатывает сигнал К СБРОС Н, производя первоначальную установку канала, а затем снимается сигнал К ПОСТН В.

4.2. Общая теория работы центрального процессора

Основным устройством ЭВМ является центральный процессор, который управляет распределением времени использования канала внешними устройствами и выполняет все необходимые арифметико-логические операции для обработки информации. Он содержит 8 быстродействующих регистров общего назначения (РОН), которые широко используются при выполнении различных операций. Центральный процессор выполняет одноадресные, двухадресные команды, команды расширенной арифметики и с плавающей запятой, может обрабатывать как шестнадцатиразрядные слова, так и 8-разрядные байты. Возможность использования восьми методов адресации позволяет вести высокоэффективную обработку данных, хранимых в любой ячейке памяти или в регистре.

4.2.I. Регистры общего назначения.

Модуль ЦП содержит 16-разрядные РОН, способные выполнять различные функции. Они могут служить в качестве накопительных регистров, индексных регистров, регистров автоинкрементной адресации, так называемых указателей стека, и для других целей. Регистры общего назначения используются для выборки операндов и записи результатов при выполнении арифметико-логических операций аналогично ячейкам памяти и регистрам внешних устройств. Два из восьми регистров общего назначения R0÷R7 имеют, кроме того, специальное назначение. Регистр R6 используется как указатель стека (УС) и содержит адрес последней заполненной ячейки стека. Регистр R7 служит счетчиком команд (СК) и содержит адрес очередной выполняемой команды. Обычно он используется только для адресации и не используется как накопительный регистр. Операции по выполнению команд с регистровым методом адресации являются внутренними по отношению к микропроцессору и не требуют выполнения обращения к каналу (за исключением цикла выборки команды). Обмен же данными с памятью и внешними устройства выполняется через канал и занимает более длительное время. Таким образом, пспользование РОН для хранения операндов при процессорных операциях повышает быстродействие ЭВМ.

4.2.2. Обращение к памяти и внешним устройствам

Весь обмен информацией и управляющими сигналами между различными устройствами ЭВМ осуществляется через единый канал передачи информации. 16-разрядный код адреса позволяет обращаться к 32К 16-разрядных ячеек. Нижние 4К адресов в ($0 \div 17777_8$) обычно предназначаются для резидентной памяти, размещенной на модуле ЦП. Верхние 4К адресов (28К ÷ 32К) отве-

дены под регистры внешних устройств. Однако, пользователь не обязан использовать все адреса этого пространства для этой цели и может руководствоваться соображениями необходимости.

Как показано на рис. I9. слово делится на старший и младший байты. Ячейки, содержащие полные слова, всегда имеют четные адреса. Адреса могут быть как четными, так и нечетными.

Младшие байты слов хранятся в ячейках с четными адресами, старшие с нечетными (рис. 20).

Рис. I9 Формат слова ЭВМ

Слово			Байт	
байт	байт			
Старший	Младший	000000	Младший	000000
Старший	Младший	000002	Старший	000001
Старший	Младший	000004	Младший	000002
Старший	Младший	000006	Старший	000003
Старший	Младший	000010	Младший	000004
Старший	Младший	017772	Старший	017775
Старший	Младший	017774	Младший	017776
Старший	Младший	017776	Старший	017777

Рис. 20 Организация памяти ЭВМ по словам и байтам для первых 4К адресов

Часть адресов зарезервирована для использования при обработке внутренних и внешних прерываний программы (адреса $0 \div 376_8$). При любом обращении к памяти или внешнему устройству используются следующие циклы обращения к каналу:

ВВОД – передача слова в процессор (эквивалентно операции ЧТЕНИЕ),

ВВОД–ПАУЗА–ВЫВОД (Б) – передача в процессор, сопровождаемая передачей слова из процессора (эквивалентно операции ЧТЕНИЕ–МОДИФИКАЦИЯ–ЗАПИСЬ) или передача слова в процессор, сопровождаемая передачей байта из процессора (эквивалентно операции ЧТЕНИЕ–МОДИФИКАЦИЯ–ЗАПИСЬ БАЙТА), ВЫВОД (Б) – передача слова из процессора (эквивалентно операции ЗАПИСЬ) или передача байта из процессора (эквивалентно операции ЗАПИСЬ БАЙТА).

Для выполнения каждой команды требуется один или несколько циклов обращения к каналу. Первым из них является ВВОД, во время которого команда извлекается из ячейки памяти с адресом, определяемым содержимым счетчика команд (СК). Если при исполнении команды не нужно обращаться за операндами к памяти или к регистрам внешних устройств, дополнительных циклов обращения к каналу не требуется. В противном случае для выборки операндов и записи результата выполняется один или несколько дополнительных циклов обращения к каналу. Следует отметить различие между операциями по прерыванию программы и операциями по прямому доступу к памяти (ПДП). В первом случае содержимое регистров процессора может быть на-

рушено, поэтому прерывание программы происходит только после окончания выполнения текущей программы. Операции по ПДП могут происходить в промежутках между циклами обращения к каналу, поскольку они не связаны с нарушением состояния процессора.

4.2.3. Слово состояния процессора (ССП)

Слово состояния процессора содержит информацию о текущем состоянии процессора. Это информация о текущем приоритете процессора, значении кодов условий, ветвления, зависящем от результата выполнения команды, и состояния Т- разряда, используемого при отладке программ и вызывающего прерывание программы. На рис. 2I показан формат ССП. Разряд приоритета процессора (7 разряд ССП) может находиться в состоянии "0" или "I". В последнем случае внешние устройства не могут вызывать прерывание текущей программы. Для удовлетворения требований прерывания программы разряд 7 ССП должен быть равен 0.

Коды условий ветвления содержат информацию о результате последней выполненной процессором команды. Процедуру установки их в соответствующее состояние выполняют все арифметические и логические одноадресные или двухадресные команды. Установка отдельных разрядов этих кодов выполняется в следующих случаях: Z = I, если результат равен 0,

N = I, если результат отрицателен,

С =I, если в результате выполнения команды произошел перенос из самого старшего разряда, или если при сдвиге вправо или влево из самого младшего или самого старшего разряда была выдвинута единица.

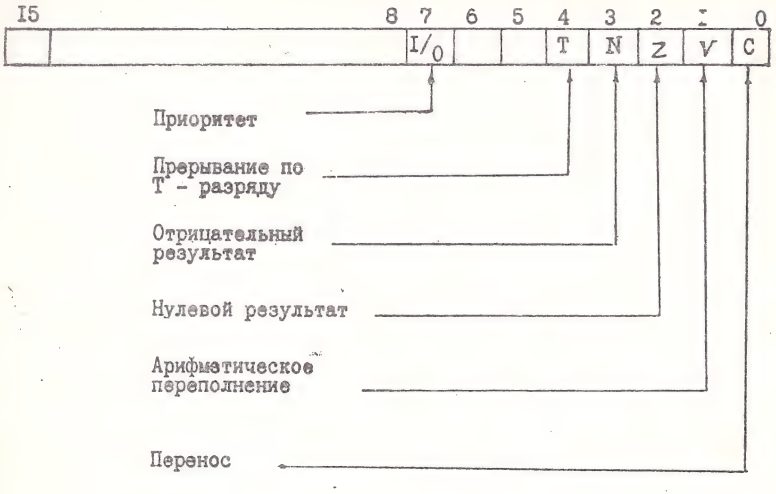


Рис. 2I Слово состояния процессора (ССП)

V= I, если в результате выполнения команды произошло арифметическое переполнение.

При выборке из стека нового слова состояния процессора и занесении его в регистр состояния процессора может устанавливаться или очищаться Т-разряд ССП. Если он установлен, то по завершении выполнения текущей команды будет вызвано прерывание программы с вектором I4 и из ячейки I6 будет занесено в регистр состояния новое слово состояния процессора. Использование Т-разряда особенно эффективно в отладочных программах для организации такого режима выполнения отлаживаемой программы, когда исполнение интересующих пользователя команд вызывает прерывание программы и переход на программу связи с оператором. Это дает возможность пользователю предпринять необходимые действия.

4.2.4. Выполнение команд

В отличие от большинства I6-разрядных ЭВМ, система команд которых состоит обычно из трех различных классов (безадресные команды, адресные команды и команды ввода/вывода), центральный процессор выполняет все эти операции одним набором команд. Поскольку процессор обращается к регистрам внешних устройств таким же способом, как и к ячейкам памяти, команды, используемые для работы с данными, находящимися в памяти, применяются и к данным, находящимся в регистрах внешних устройств. Например, можно анализировать данные, находящиеся в регистрах внешних устройств, не пересылая их в память или РОН, или выполнять над ними арифметические и логические преобразования.

Центральный процессор выполняет одноадресные и двухадресные команды как с 16-разрядными словами, так и с 8-разрядными байтами. Поэтому очень эффективно могут быть выполнены такие операции, как сложение или вычитание двух операндов или пересылка операнда из одной ячейки в другую. Ниже сравниваются традиционный подход к выполнению таких операций и подход, принятый в ЦП.

Традиционный подход

ЗАГРУЗИТЬ А Загрузить содержимое ячейки памяти А в накопительный регистр.

СЛОЖИТЬ Б Сложить содержимое ячейки Б с содержимым накопительного регистра.

ЗАПИСАТЬ Б Записать результат в ячейку Б.

Подход, принятый в ЦП.

СЛОЖИТЬ А,Б Содержимое А складывается с содержимым ячейки Б и результат записывается в ячейку Б.

Богатые вычислительные возможности ЦП во многом определяются широким выбором методов адресации. Они позволяют осуществлять последовательную адресацию в прямом и обратном направлениях, индексную адресацию, косвенную адресацию, адресацию с использованием памяти, организованной по принципу стека. Переменный формат команд позволяет использовать минимальное количество слов для каждого метода адресации, что дает в результате экономию памяти.

Выполнение команды начинается с операции по выборке команды. Процессор извлекает из ячейки памяти с заданным адресом

код команды, используя для этого цикл ВВОД. После дешифрации команды процессор начинает операцию по выборке операндов, инициируя, если это необходимо, один или несколько циклов ВВОД, в зависимости от кода команды и метода адресации. Далее производится операция по исполнению команды и, если требуется записать результат в память или регистр ВУ, начинается цикл ВЫВОД (или заканчивается цикл ВВОД-ПАУЗА-ВЫВОД; который мог начаться при выборке операнда приемника). Затем процессор может начинать операцию обслуживания, которая вызывается выполнением внутренних или внешних прерываний. Для различных команд последовательность операций может быть различной. Но любую из них ЦП начинает с операции по выборке команды, переходя затем к выполнению какой-либо другой операции, или к их последовательности. В некоторых случаях после операции выборки текущей команды, процессор может перейти к операции выборки следующей команды.

4.3. Система команд

В центральном процессоре ЭВМ используются три типа команд: безадресные, одноадресные и двухадресные. В безадресных командах код команды содержит только код операции. В кодах одноадресных и двухадресных команд обычно содержится информация, которая определяет:

- выполняемую функцию (код операции),
- регистры общего назначения, используемые при выборке операндов,
- метод адресации (способ использования выбранного РОН).

Так как большая часть информации обрабатываемой вычислительной машиной, представляет собой данные, сформированные в массивы, списки, потоки символов и т.д., центральный процессор разработан с учетом возможности эффективной обработки сформированных структур данных. Регистры общего назначения могут быть использованы:

– как накопители (обрабатываемые данные хранятся в регистрах),

– как указатели адреса (регистр общего назначения содержит адрес операнда, а не сам операнд),

– как указатели адреса, содержимое которых изменяется автоматически с заданным шагом, что позволяет обращаться к последовательно расположенным ячейкам памяти. Автоматическое увеличение содержимого указателя при обращении к последовательно расположенным ячейкам памяти носит название автоинкрементной адресации. Автоматическое уменьшение содержимого указателя при обращении к последовательно расположенным ячейкам памяти носит название автодекрементной адресации. Эти методы могут быть использованы при обработке табулированных данных;

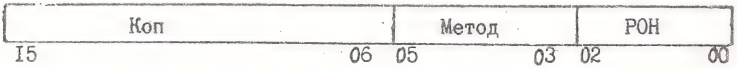
– как индексные регистры, содержимое которых приравнивается к индексному слову для вычисления адреса операнда. Это позволяет легко обращаться к различным элементам списка.

Использование автоинкрементного и автодекрементного методов адресации дает возможность организации стековой памяти. В качестве указателя стека программно может быть выбран любой РОН, однако определенные команды (используемые при обслуживании прерываний, обращении и возврате из подпрограмм)

автоматически используют РОН R6 как аппаратный указатель стека.

4.3.I. Формат одноадресных команд

Формат команды для всех одноадресных команд (таких как очистка, проверка) имеет следующий вид:



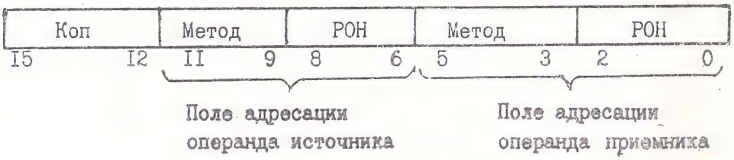
Разряды I5 ÷ 06 содержат код операции, который определяет выполняемую команду. Разряды 05 ÷ 00 образуют шестиразрядное поле, именуемое полем адресации операнда приемника, которое в свою очередь состоит из двух подполей:

а) разряды 02 ÷ 00 определяют один из восьми РОН, который использует данная команда,

б) разряды 05 ÷ 03 определяют способ использования выбранного регистра (метод адресации). Причем разряд 03 определяет прямую или косвенную адресации.

4.3.2. Формат двухадресных команд.

Операции над двумя операндами (такие, как сложение, пересылка, сравнение) выполняются с помощью команд, в которых имеются два адреса. Задание разрядов в поле адресации первого и второго операндов определяет различные методы и различные регистры общего назначения. Формат двухадресной команды имеет следующий вид:



Поле адресации операнда источника используется для выборки операнда источника. Поле адресации операнда приемника используется для выборки операнда приемника и занесения результата. Например, по команде ADD A,B складывается содержимое ячейки A (операнд источника) с содержимым ячейки B (операнд приемника). После выполнения операции сложения в ячейке B будет находиться результат операции, а содержимое ячейки A не изменится.

4.3.3. Методы адресации.

В примерах данного подраздела при описании методов адресации используются следующие команды:

Таблица 5

Мнемоника	Описание	Восьмеричный код
CLR	Очистка. Очищается указанная ячейка	0050DD
CLRB	Очистка байта. Очищается старший или младший байт указанной ячейки	1050DD
INC	Прибавление единицы. К содержимому указанной ячейки прибавляется единица	0052DD
INCB	Прибавление единицы к байту	1052DD
COM	Инвертирование. Заменяет первый операнд его обратным кодом, т.е. каждый разряд, равный 0, уста-	0051DD

Мнемоника	Описание	Восьмеричный код
	навливается, а каждый разряд, равный I, очищается.	
COMB	Инвертирование байта	I05IDD
ADD	Сложение. Содержимое второго операнда складывается с содержимым первого операнда и результат заносится по адресу первого операнда	0655DD

4.3.3.I. Методы прямой адресации

На рис. 22 показаны последовательности операции выполнения команды с каждым из четырех методов прямой адресации. При регистровом методе адресации любой из восьми регистров общего назначения может быть использован как накопитель. Следовательно, операнд будет находиться в выбранном регистре. Так как РОН аппаратно реализован в БИС РАЛУ, они обладают более высоким быстродействием, чем любая память, работающая под управлением процессора. Это преимущество особенно проявляется при операциях с переменными, к которым необходимо часто обращаться.

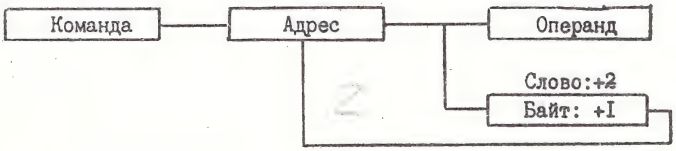
Примеры регистровой адресации

Символическое обозначение	Восьмеричный код	Наименование команды
I. INC R3	005203	Прибавление единицы

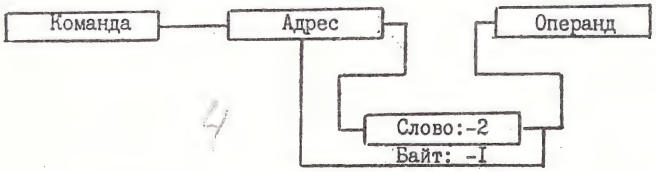
Действие: Прибавляет единицу к содержимому третьего РОН

Регистровый метод адресации

Автоинкрементный метод адресации



Автоинкрементный метод адресации



Индексный метод адресации

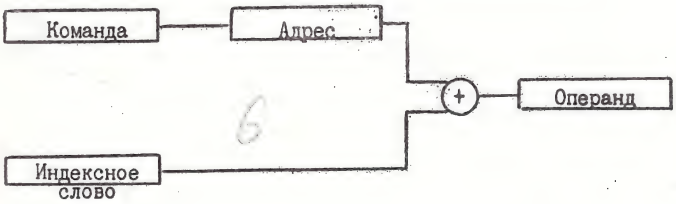
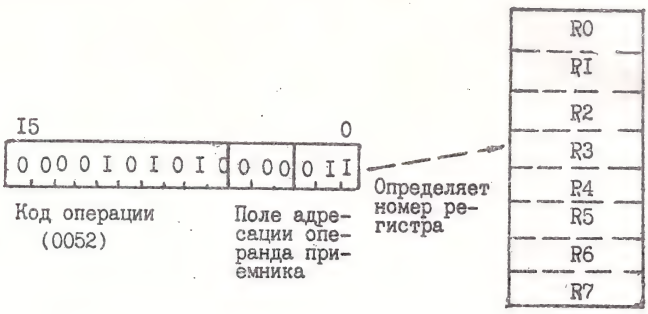


Рис.22 Методы прямой адресации



2. ADD R2, R4 060204 Сложение

Действие: складывается содержимое R2 с содержимым R4.

До выполнения операции:	После выполнения операции
R2) 000002	R2) 000002
R4) 000004	R4) 000006

3. COMB R4 I05I04 Инвертирование байта

Действие: Заменяется содержимое разрядов 07 ÷ 00 в R4 их обратным кодом.(Байтовые команды, использующие регистры общего назначения, оперируют только с младшим байтом информации, содержащимся в РОН).

До выполнения операции	После выполнения операции
R4) 022222	R4) 022I55

Автоинкрементный метод адресации используется при автоматическом обращении с заданным шагом к последовательным элементам таблицы операндов. Содержимое выбранного регистра автоматически наращивается для возможности обращения в даль-

нейшем к последующей ячейке. При байтовых операциях наращивание происходит на единицу, при операциях с полными словами – на 2, а содержимое R6, R7 всегда наращивается на 2. Автоинкрементный метод адресации особенно удобен при операциях с массивами и стеками. С помощью этого метода можно выбрать элемент таблицы, а затем нарастить указатель для обращения к следующему операнду в таблице. Хотя этот метод наиболее удобен при работе с таблицами, он может быть использован как общий метод для различных целей.

Примеры автоинкрементной адресации; (2)

I. CLR (R5)+ 005025 Очистка

Действие: Содержимое ячейки, адрес которой содержится в R5, очищается, после чего адрес (содержимое R5) увеличивается на 2.

До выполнения операции:

20000) 005025 R5) 030000
30000) IIIII6

После выполнения операции:

20000)005025 R5)030002
30000) 000000

2. CLRB (R5)+ I05025 Очистка байта

Действие: Очищается выбранный байт, адрес которого содержится в R5, после чего адрес (содержимое R5) увеличивается на единицу.

До выполнения операции:

20000) 105025 R5) 030000
30000) IIIII6
30002)

После выполнения операции:

20000) I05025 R5)030001
30000) III000
30002)

3. ADD (R2)+R4 062204 Сложение

Действие: Операнд, адрес которого содержится в R2, складывается с содержимым R4. Результат заносится в R4, а содержимое R2 (адрес операнда источника) наращивается на 2.

До выполнения операции:	После выполнения операции:
I00) 062204 R2) 000204	I00) 062204 R2) 000206
R4) 001000	R4) 003000
204) 002000	204) 002000

Автодекрементный метод адресации также используется для обработки табулированных данных. Однако, в отличие от автоинкрементного метода, адресация к ячейкам массива идет в противоположном направлении. При этом методе адресации содержимое выбранного РОН уменьшается (для байтовых команд на единицу, для команд с полными словами – на два), а затем используется как исполнительный адрес.

Сочетание автоинкрементного и автодекрементного методов адресации может быть эффективно использовано для обработки стека.

Примеры автодекрементной адресации: (4)

Символическое обозначение	Восьмеричный код	Наименование команды
I. INC -(R0)	005240	Прибавление единицы

Действие: Содержимое R0 уменьшается на 2 и используется как исполнительный адрес. К содержимому, выбранному из ячейки по этому адресу, прибавляется единица.

До выполнения операции:	После выполнения операции:
I00) 005240 R0) 0I7776	I00)005240 R0)0I7774
I7774) 000000	I7774) 00000I

2. *INC B - (R0)* I05240 Прибавление единицы к байту

Действие: Содержимое R0 уменьшается на единицу и используется как адрес операнда. К выбранному байту прибавляется единица.

До выполнения операции:	После выполнения операции:
I00) I05240 R0) 0I7776	I00) I05240 R0) 0I7775
I7774) 000000	I7774) 000400

3. *ADD - (R3), R0* 064300 Сложение

Действие: Содержимое R3 уменьшается на 2 и используется как адрес операнда источника, который складывается с содержимым R0 (операнд приемника). Результат записывается в R0.

До выполнения операции:	После выполнения операции:
I20) 064300 R0) 000020	I20) 064300 R0) 000070
R3) 00I776	R3) 00I774
I774)000050	I774) 000050

Индексный метод адресации, при котором исполнительный адрес определяется как сумма содержимого выбранного РОН с индексным словом позволяет осуществлять произвольный доступ к элементам структуры данных. Индексное слово содержит-

ся в следующей за командным словом ячейке памяти. При индексном методе адресации содержимое выбранного регистра может быть использовано в качестве базы для вычисления серии адресов.

Примеры индексной адресации: (6)

Символическое обозначение	Восьмеричный код	Наименование команды
I. CLR 200(R4)	005064 000200	Очистка

Действие: адрес операнда определяется прибавлением к содержимому R4 кода 200, после чего ячейка с вычисленным адресом очищается.

До выполнения операции:	После выполнения операции:
I020) 005064 R4)00I000	I020) 005064 R4) 00I000
I022) 000200	I022) 000200
I200) I77777	I200) 000000

2. COMB 200 (RI)	I05I6I 000200	Инвертирование байта

Действие: адрес операнда определяется прибавлением к содержимому RI кода 200. После чего содержимое ячейки с вычисленным адресом заменяется его обратным кодом.

До выполнения операции:	После выполнения операции:
I020) I05I6I RI) 0I7777	I020) I05I6I RI) 0I7777
I022) 000200	I022) 000200
20I76) 0II000	20I76) I66400

3.858.382 ТО

Адрес операнда был вычислен прибавлением к содержимому R1 кода 200, в результате чего команда COMB произвела свое действие над старшим байтом ячейки с адресом 20I76.

3. ADD 30 (R2), 20 (R5) 066265 Сложение
000030
000020

Действие: Содержимое ячейки, адрес которой вычисляется сложением кода 30 с содержимым R2 (операнд источника), складывается с содержимым ячейки, адрес которой вычисляется сложением кода 20 с содержимым R5 (операнд приемника). Результат записывается по адресу операнда приемника.

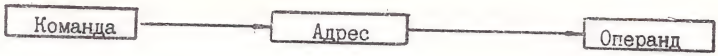
До выполнения операции:	После выполнения операции:
I020) 066265 R2) 00II00	I020) 066265 R2) 00II00
I022) 000030 R5) 002000	I022) 000030 R5) 002000
I024) 000020	I024) 000020
II30) 00000I	II30) 00000I
2020) 00000I	2020) 000002

4.3.3.2. Методы косвенной адресации

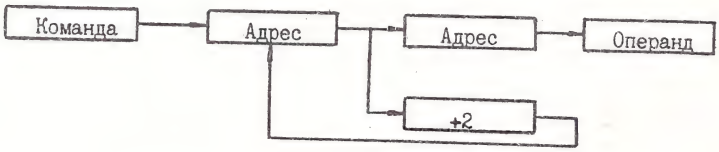
Четыре основных метода могут быть использованы с косвенной адресацией. Если в регистровом методе операндом является содержимое выбранного регистра, в косвенно-регистровом методе содержимое выбранного регистра является адресом операнда. В трех других косвенных методах содержимое регистра позволяет выбрать адрес операнда, а не сам операнд. Эти методы используются, когда таблица состоит из адресов, а не из операндов.

На рис. 22 показаны последовательности операций выпол-

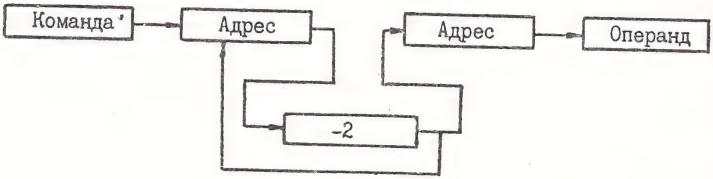
Косвенно-регистровый метод адресации



Косвенно-автоинкрементный метод адресации



Косвенно-автодекрементный метод адресации



Косвенно-индексный метод адресации

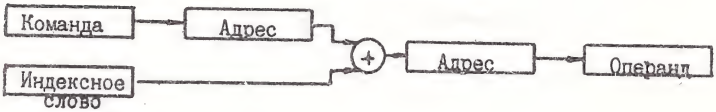


Рис. 23 Методы косвенной адресации.

нения команды с каждым из четырех методов косвенной адресации.

Пример косвенно-регистрового метода адресации.

Символическое обозначение	Восьмеричный код	Наименование команды
CLR * R5	005015	Очистка

Действие: Содержимое ячейки, адрес которой содержится в R5, очищается.

До выполнения операции:	После выполнения операции:
1020) 005015 R5) 001700	1020) 005015 R5) 001700
1700) 177777	1700) 000000

Пример косвенно-автоинкрементного метода адресации

Символическое обозначение	Восьмеричный код	Наименование команды
INC * (R2)+	005232	Прибавление единицы

Действие: Содержимое ячейки, адрес которой содержится в R2, используется как адрес операнда. Операнд увеличивается на единицу, а содержимое R2 на два.

До выполнения операции:	После выполнения операции:
1000) 005232 R2) 010300	1000) 005232 R2) 010302
1010) 000000	1010) 000001
10300) 001010	10300) 001010

Пример косвенно-автодекрементной адресации

Символическое обозначение	Восьмеричный код	Наименование команды
COM * - (R0)	005150	Инвертирование

Действие: Содержимое R0 уменьшается на два и используется как адрес ячейки, в которой находится адрес операнда. Операнд заменяется его обратным кодом.

До выполнения операции:	После выполнения операции:
I000) 005I50 R0) 0I0776	I000) 005I50 R0) 0I0774
I0I00) 0I2345	I0I00) I65432
I0774) 0I0I00	I0774) 0I0I00

Пример косвенно-индексной адресации

Символическое обозначение	Восьмеричный код	Наименование команды
ADD @ I000 (R2), R1	06720I 00I000	Сложение

Действие: Содержимое ячейки, адрес которой определяется сложением кода I000 с содержимым R2, складывается с операндом, хранящимся в R1. Результат записывается в R1.

До выполнения операции:	После выполнения операции:
I020) 06720I R1) 00I234	I020) 06720I R1) 00I236
I022) 00I000 R2) 000I00	I022) 00I000 R2) 000I00
I050) 000002	I050) 000002
II00) 00I050	II00) 00I050

4.3.3.3. Использование счетчика команд (СК) в качестве регистра общего назначения.

Регистр R7, являясь одним из восьми РОН, выполняет в центральном процессоре специальную функцию счетчика команд.

Когда центральный процессор использует счетчик команд для выборки слова из памяти, его содержимое автоматически увеличивается на 2. Новое содержимое счетчика команд является адресом слова, используемого при выполнении данной команды или адресом следующей команды, которую необходимо выполнить. Следует отметить, что при работе с байтами содержимое СК также увеличивается на 2.

Счетчик команд может быть использован во всех методах адресации, применяемых ЦП. Однако, наиболее эффективно он использ- уется только с четырьмя методами адресации. Эти методы адресации получили специальные наименования: непосредственный, абсолютный, относительный и косвенно-относительный. Использование этих методов дает возможность построения программы, работоспособность которой не теряется при перемещении ее в любую область памяти. В таблице 4 приведены методы адресации с использованием R7.

Необходимо помнить, что эти четыре метода такие же, как и методы, описанные в пп. 4.3.3.1 и 4.3.3.2, но в качестве РОН используется R7.

Таблица 6

Двоичный код	Наименование	Функция
010	Непосредственный	Операнд выбирается из ячейки, следующей за командным словом.
011	Абсолютный	Из ячейки, следующей за командным словом, выбирается адрес операнда.

Продолжение табл. 6

Двоичный код	Наименование	Функции
110	Относительный	Операнд выбирается из ячейки, адрес которой определяется как сумма содержимого СК и ячейки, следующей за командным словом.
111	Косвенно-относительный	Из ячейки, адрес которой определяется как сумма содержимого СК и ячейки, следующей за командным словом, выбирается адрес операнда

Методы адресации с использованием СК в значительной мере упрощают обработку данных, не сформированных в массивы.

Непосредственный метод адресации имеет символическое обозначение # ⊓ . Он эквивалентен автоинкрементному методу адресации через СК.

Этот метод обеспечивает удобство написания программы и экономию времени программиста путем помещения константы в ячейку памяти вслед за командным словом.

Пример непосредственной адресации

Символическое обозначение	Восьмеричный код	Наименование команды
ADD # 10 , R0	062700	Сложение
	000010	

Действие: Содержимое R0 складывается с числом 10. Результат записывается в R0.

До выполнения операции:	После выполнения операции:
I020) 062700 R0) 000020	I020) 062700 R0) 000030
I022) 000010	I022) 000010

П р и м е ч а н и е: Непосредственно перед выборкой этой команды СК указывает на командное слово. ЦП выбирает командное слово и увеличивает СК на 2. В поле адреса операнда источника записан код 27, следовательно, СК используется как указатель при выборке операнда перед увеличением его содержимого на 2 для указания на следующую команду.

Абсолютный метод адресации имеет символическое обозначение $*\#A$. Он эквивалентен косвенно-автоинкрементной адресации через СК. Этот метод удобен тем, что адрес операнда является его абсолютным адресом (т.е. он остается постоянным независимо от места расположения программы в памяти).

Пример абсолютной адресации.

Символическое обозначение	Восьмеричный код	Наименование команды
CLR * # II00	005037	Очистка
	00II00	

Действие: Содержимое ячейки, следующей за командой, используется в качестве адреса операнда (в данном случае исполнительным адресом является код II00). Содержимое ячейки II00 очищается.

До выполнения операции:	После выполнения операции:
20) 005037	20) 005037
22) 00II00	22) 00II00
II00) I7777	II00) 000000

Относительный метод адресации имеет символическое обозначение X (СК) или A, где X - исполнительный адрес по отношению к счетчику команд. Этот метод эквивалентен индексной адресации через СК. Индексное слово хранится в следующей за командным словом ячейке и, будучи сложено с содержимым СК, дает адрес операнда. Этот метод полезен при написании программы, которая может располагаться в различных местах памяти, так как адрес операнда фиксируется по отношению к содержимому СК. При необходимости перемещения программы в памяти, операнд перемещается на то же число ячеек, что и сама команда.

Пример относительной адресации

Символическое обозначение	Восьмеричный код	Наименование команды
I N CA	005267	Прибавление
	000054	единицы

Действие: К операнду, адрес которого определяется сложением содержимого СК и индексного слова (000054), прибавляется единица.

До выполнения операции:	После выполнения операции:
I020) 005267	I020) 005267
I022) 000054	I022) 000054
I024)	I024)
II00) 000000	II00) 000001

Косвенно-относительный метод адресации имеет символическое обозначение $*X(СК)$ или $*A$, где X – адрес ячейки, содержащей исполнительный адрес, по отношению к счетчику команд. Этот метод эквивалентен косвенно-индексной адресации через СК.

Пример косвенно-относительной адресации

Символическое обозначение	Восьмеричный код	Наименование команды
CLR * A	005077	Очистка
	000020	

Действие: Очищает содержимое ячейки, адрес которой определяется следующим образом: к содержимому счетчика команд прибавляется индексное слово, определяя тем самым адрес ячейки, в которой хранится исполнительный адрес.

До выполнения операции:	После выполнения операции:
I020) 005077	I020) 005077
I022) 000020	I022) 000020
I024)	I024
I044) 0I0I00	I044) 0I0I00
I0I00) I23456	I0I00) 000000

4.3.3.4. Использование указателя стека в качестве регистра

общего назначения.

Регистр R6, являясь одним из РОН, используется в ЦП как указатель адреса при обращении к той части памяти, которая отводится под стек. С помощью автодекрементной адресации через R6 данные записываются в стек, а с помощью автоинкрементной адресации производится выборка данных из стека. Индексный метод адресации позволяет производить произвольную выборку элементов стека.

Так как R6 (УС) используется для обслуживания прерываний, то его особенностью является то, что уменьшение и увеличение содержимого УС всегда производится с шагом на 2. В байтовых операциях, в которых используется УС, содержимое ячеек с нечетными адресами не изменяется.

4.3.4. Выполнение команд

Описание каждой команды включает: мнемонику, восьмеричный код, формат команды, двоичный код, описание выполнения команды и выработки признаков, специальные пояснения и примеры.

При описании команд используются следующие обозначения:

R – регистр общего назначения;

СК – счетчик команд (R7);

УС – указатель стека (R6);

РСП – регистр состояния процессора;

ССП – слово состояния процессора;

SS – поле адресации операнда источника;

src – источник;

(src) – операнд источника;

DD – поле адресации операнда приемника;

dst – приемник;

(dst) – операнд приемника;

XXX – смещение (8 разрядов);

NN – смещение (6 разрядов);

( ) – содержимое ячейки;

$\wedge$ – логическое умножение ("И");

$\vee$ – логическое сложение ("ИЛИ");

$\forall$ – "исключающее ИЛИ";

$\overline{A}$ – отрицание A ("НЕ");

$\leftarrow$ – становится равным;

$\downarrow$ – запись в стек;

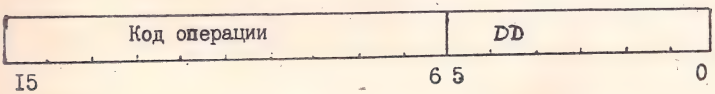
$\uparrow$ – выборка из стека;

B – байтовая команда.

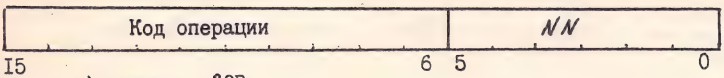
4.3.4.I. Форматы команд

Ниже приведены форматы команд, реализуемые ЦП.

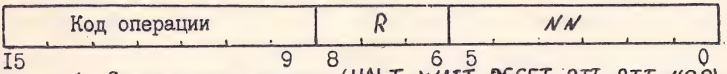
I. Одноадресные команды (CLR, COM, INC, DEC, NEG, ADC, SBC, TST, ROR, ROL, ASR, ASL, JMP, SWAB, MEPS, MFPS, MTPS, SXT)



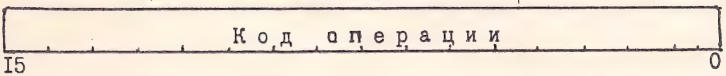
д) команда MARK



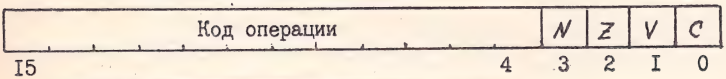
е) команда SOB



4. Специальные команды (HALT, WAIT, RESET, RTI, RTT, NOP)



5. Команды изменения признаков



4.3.4.2. Выполнение байтовых команд

Большинство команд ЦП оперируют как с полными словами, так и с байтами. Байтовые команды с автоинкрементным или автодекрементным методами адресации изменяют содержимое указанного регистра на "I" для обращения к следующему байту. Байтовые команды при регистровом методе адресации производят обработку младшего байта выбранного регистра. Если старший разряд командного слова (разряд I5) установлен, он указывает, что команда байтовая. Если же в разряде I5 командного слова записан "0", команда оперирует с полным словом.

Пример:

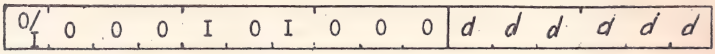
CLR	0050DD	Очистка слов
CLR B	I050DD	Очистка байта

4.3.4.3. Одноадресные команды

CLR 0050DD

CLR B I050DD

Очистка



Действие: $(dst) \leftarrow 0$

Признаки:

N – очищается

Z – устанавливается

V – очищается

C – очищается

Описание: В указанную ячейку записываются нули. Для байтовой команды нули записываются в указанный байт.

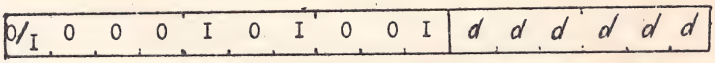
Пример: CLR RI

До выполнения операции:	После выполнения операции:
(RI) = I77777	(RI) = 000000
NZVC IIII	NZVC 0I00

COM 005IDD

COMB I05IDD

Инвертирование



Действие: $(dst) \leftarrow \overline{(dst)}$

Признаки: N – устанавливается, если результат < 0,
в противном случае очищается;
Z – устанавливается, если результат $= 0$,
в противном случае очищается;
V – очищается;
C – устанавливается.

Описание: Заменяет содержание указанной ячейки его двоичным обратным кодом (каждый разряд, содержащий "0", устанавливается, а каждый разряд, содержащий "I", очищается). Для байтовой команды операция производится по отношению к указанному байту.

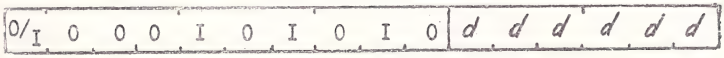
Пример: COM R0

До операции	После операции
(R0) = 013333	(R0) = 164444
NZVC 0110	NZVC 1001

INC 005200

INCB 105200

Прибавление единицы



Действие: $(dst) \leftarrow (dst)+1$

Признаки: N – устанавливается, если результат < 0, в противном случае очищается;

Z – устанавливается, если результат = 0, в противном случае очищается,

V – устанавливается, если операнд равен 077777, в противном случае очищается,

C – не изменяется.

Описание: Прибавляет единицу к содержимому указанной ячейки (или байту, если команда байтовая).

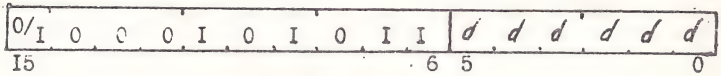
Пример: INC R2

До операции:	После операции
(R2) = 000333	(R2) = 000334
NZVC 0000	NZVC 0000

DEC
DECB

0053DD
1053DD

Вычитание единицы



Действие: (dst) ← (dst) – 1

Признаки: N – устанавливается, если результат < 0, в противном случае очищается;

Z – устанавливается, если результат = 0, в противном случае очищается;

V – устанавливается, если операнд был равен 100000, в противном случае очищается;

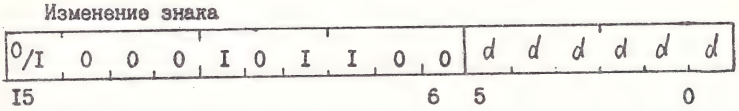
C – не изменяется.

Описание: Из содержимого указанной ячейки (или указанного байта для байтовых команд) вычитается единица.

Пример: DEC R5

До операции	После операции
(R5) = 00000I	(R5) = 000000
NZVC	NZVC
I000	0I00
NEG	0054DD
NEGB	I054DD

Изменение знака



Действие: (dst) ← (dst)

Признаки: N – устанавливается, если результат < 0, в противном случае очищается;

Z – устанавливается, если результат = 0, в противном случае очищается;

V – устанавливается, если результат = I00000, в противном случае очищается;

C – очищается, если результат = 0, в противном случае устанавливается.

Описание: Содержимое указанной ячейки (или байта для байтовых команд) заменяется двоичным дополнением операнда. Следует заметить, что число I00000 заменяется самим собой, так как не существует соответствующего ему положительного числа.

Пример: NEG R0

До операции: После операции:

(R0) = 0000I0	(R0) = I77770
NZVC	NZVC
0000	I00I
TST	0057DD
TSTB	I057DD

Проверка:

Действие: (dst) ← (dst)

Признаки: N - устанавливается, если результат < 0, в противном случае очищается;

Z - устанавливается, если содержимое = 0, в противном случае очищается;

V - очищается;

C - очищается.

Описание: В зависимости от содержимого указанной ячейки (или байта для байтовых команд) устанавливается или очищается признаки N и Z.

Пример: TST RI

До операции:	После операции:
(RI) = 0I2340	(RI) = 0I2340
NZVC	NZVC
00II	0000

3.858.382 ТО

Увеличение или уменьшение чисел в кратное степени 2 число раз выполняется с помощью команд арифметического сдвига:

ASR – арифметического сдвига вправо и *ASL* – арифметического сдвига влево. Знаковый разряд операнда (разряд I5) при арифметическом сдвиге вправо восстанавливается. В младший разряд при арифметическом сдвиге влево заносится нуль. Информация, сдвинутая за пределы С – разряда теряется.

ASR 0062DD

ASRB I062DD

Арифметический сдвиг вправо

Действие: *(dst)* – сдвинутое на одну позицию вправо (*dst*)

Признаки: *N* – устанавливается, если старший разряд результата установлен, в противном случае очищается;

Z – устанавливается, если результат = 0, в противном случае очищается,

V – устанавливается результатом операции "исключающее ИЛИ" над содержимым N и С разрядов, которые они имеют после операции сдвига,

C – загружается содержимым младшего разряда указанной ячейки.

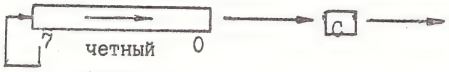
Описание: Сдвигаются все разряды операнда вправо на одну позицию. Содержимое знакового разряда восстанавливается. C – разряд загружается содержимым младшего разряда операнда.

Таким образом, ASR или ASRB выполняет деление числа со знаком на два.

Пример:

Сдвиг полного слова

Сдвиг байта



ASL 0063DD

ASLB I063DD

Арифметический сдвиг влево

Действие. (dst) ← сдвинутое на одну позицию влево (dst)

Признаки: N – устанавливается, если результат < 0, в противном случае очищается,

Z – устанавливается, если результат = 0, в противном случае очищается,

V – загружается результатом операции N⊻C (значения N и C после операции сдвига)

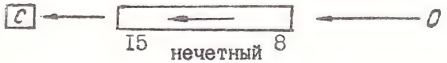
C – загружается содержимым старшего разряда операнда.

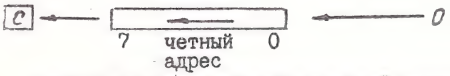
Описание: Сдвигаются все разряды операнда на одну позицию влево. В младший разряд операнда записывается нуль. С-разряд загружается содержимым старшего разряда операнда. Таким образом, ASL или ASLB выполняет умножение числа со знаком на 2.

Пример:

Сдвиг полного слова

Сдвиг байта



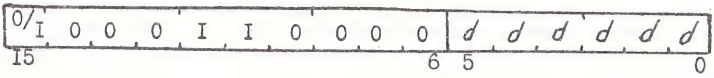


Для облегчения последовательной проверки и поразрядной обработки операнда используются команды циклического сдвига. Они оперируют со словом операнда и С-разрядом так, как если бы они образовали 17-разрядный регистр с циклическим переносом.

ROR 0060DD

RORB I060DD

Циклический сдвиг вправо



Действие: (dst) ← циклически сдвинутое на одну позицию вправо (dst);

Признаки: N - устанавливается, если результат < 0, в противном случае очищается;

Z - устанавливается, если результат = 0, в противном случае очищается;

V - загружается результатом операции N ⊻ C (значения N и C после окончания операции сдвига);

C - загружается младшим разрядом операнда.

Описание: Циклически сдвигает все разряды операнда на одну позицию вправо. Содержимое младшего разряда загружается в С-разряд, а прежнее содержимое С-разряда загружается в старший разряд операнда.

Пример:

Сдвиг полного слова

Сдвиг байта

нечетный адрес

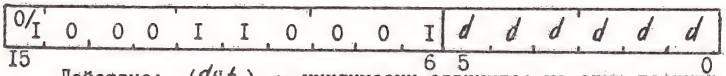
четный адрес

3.858.382 ТО

ROL 006IDD

ROLB 106IDD

Циклический сдвиг влево



Действие: (dst) ← циклически сдвинутое на одну позицию влево (dst)

Признаки: N – устанавливается, если результат < 0, в противном случае очищается:

Z – устанавливается, если результат = 0, в противном случае очищается;

V – загружается результатом операции N⊽C (значения N и C после окончания операции сдвига)

C – загружается старшим разрядом операнда.

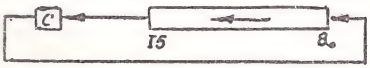
Описание: Циклически сдвигаются все разряды операнда на одну позицию влево.

Содержимое старшего разряда загружается в С-разряд, а прежнее содержимое С-разряда загружается в младший разряд операнда.

Пример:

Сдвиг полного слова

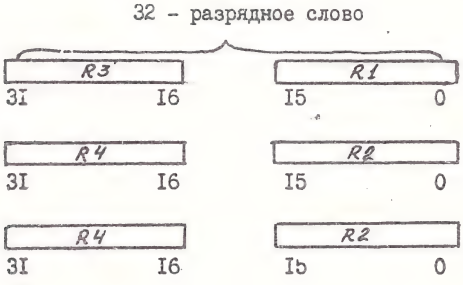
Сдвиг байта



нечетный адрес

четный адрес

При вычислениях с повышенной точностью иногда необходимо выполнять арифметические операции над операндами длиною в несколько слов. Для реализации арифметических операций с удвоенной точностью применяются команды прибавления переноса (ADC) и вычитания переноса (SBC). Например, два I6-разрядных слова могут быть объединены в 32-разрядное слово и над ним могут быть выполнены операции сложения или вычитания, как показано ниже.



Пример: Сложение – I с – I может быть выполнено следующим образом:

```
– I = 37777 777 777
  (RI), (R2), (R3), (R4) = I77777
ADD RI, R2
ADC R3
ADD R3 ,  R4
```

I. После сложения (RI) с (R2), в С-разряд загружается I.

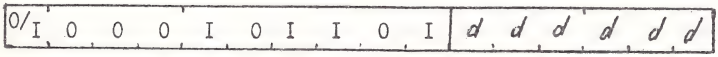
2. Команда ADC прибавляет С-разряд к (R3), после чего (R3) = 000000.

3. Складываются (R3) и (R4).

4. Результат = 37777777776 = - 2.

ADC	0055DD
ADCB	I055DD

Прибавление переноса



Действие: $(dst) \leftarrow (dst) + (C)$

Признаки:

N - устанавливается, если результат < 0, в противном случае очищается;

Z - устанавливается, если результат $= 0$, в противном случае очищается;

V - устанавливается, если перед выполнением операции (dst) = 077777, а (C) = I, в противном случае очищается;

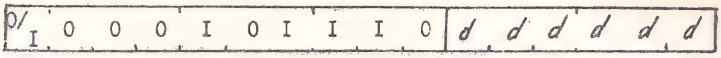
C - устанавливается, если перед выполнением операции (dst) = I77777, а (C) = I, в противном случае очищается.

Пример: ADC R3

До операции:	После операции:
(R3) = 077777	(R3) = I00000
NZVC	NZVC
000I	IOI0

SBC 0056DD
SBCB I056DD

Вычитание переноса



Действие: (dst) ← (dst) - (C)

Признаки: N – устанавливается, если результат < 0, в противном случае очищается;

Z – устанавливается, если результат = 0, в противном случае очищается;

V – устанавливается, если перед выполнением команды (dst) = I00000, в противном случае очищается,

C – устанавливается, если (dst) = 0 и (C) = I, в противном случае очищается.

Описание: Вычитается содержимое С-разряда из операнда. Это позволяет вычесть перенос, получившийся при вычитании двух младших I6-разрядных слов, из результата вычитания двух старших I6-разрядных слов.

Пример: Вычитание с удвоенной точностью осуществляется следующей последовательностью команд:

```
SUB   A0,   B0
SBC   BI
SUB   AI,   BI
```

SXT 0067DD

Расширение знака

Действие: (dst)←0, если N очищен,

(dst)←I, если N установлен

Признаки: N - не изменяется;

Z - устанавливается, если (N) = 0,

V - очищается,

C - не изменяется.

Описание: Если (N) = I, операнд заменяется кодом I77777.

Если (N) = 0, указанная ячейка очищается. Эта команда обычно используется при выполнении операций с повышенной точностью для расширения знака в 32-разрядном слове,

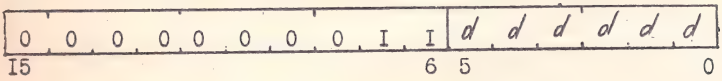
Пример: SXT R3

До операции:	После операции:
(R3) = 0I2345	(R3) = I77777
NZVC	NZVC
I000	I000

При обработке данных, представленных 8-разрядными байтами, может быть весьма полезна команда перестановки байтов (SWAB).

SWAB 0003DD

Перестановка байтов



Действие: Байт I/ Байт 0 ← Байт 0 / Байт I

Признаки: N - устанавливается, если старший разряд младшего байта установлен (разряд 7), в противном случае очищается;

Z - устанавливается, если младший байт результата = 0, в противном случае очищается;

V - очищается;

C - очищается.

Описание: Меняет местами старший байт с младшим байтом указанной ячейки. Адресация происходит по полному слову.

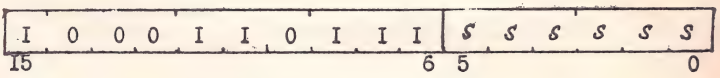
Пример: SWAB R1

До операции:	После операции:
(R1) = 077777	(R1) = 177577
NZVC	NZVC
1111	0000

Слово состояния процессора (ССП) хранится в регистре, который не имеет адреса. Возможность программного доступа к ССП обеспечивают команды чтения и записи ССП.

MFPS 1067DD

Чтение ССП



Действие: (dst)←ССП

Признаки: N - устанавливается, если разряд 7 слова состояния процессора равен 1, в противном случае очищается;

Z - устанавливается, если все восемь разрядов ССП (разряды 0-7) = 0, в противном случае очищается;

V - очищается,

C - не изменяется.

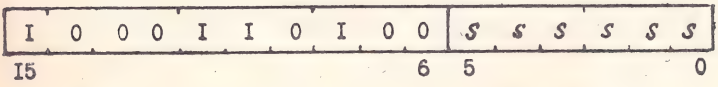
Описание: Восемь разрядов слова состояния процессора ССП переслаются в указанную ячейку. Если при этом используется регистровый метод адресации, в указанном регистре происходит расширение знака. Адрес операнда приемника воспринимается как адрес байта.

Пример: MFPS R0

До операции:	После операции:
(Ro)= 000000	(Ro) = 000014
РСП = 000014	РСП = 000000

МТРS 106455

Запись ССП



Действие: ССП — (src)

Признаки: Устанавливаются или очищаются в соответствии с разрядами 0 ÷ 3 операнда источника (src).

Описание: Восемь разрядов указанного операнда замещают содержимое слова состояния процессора. Адрес операнда источника воспринимается как адрес байта. Следует заметить, что Т-разряд (разряд 4 ССП) не может быть установлен этой командой. Операнд источника (src) не изменяет своего содержимого.

Эта команда может быть использована для изменения приоритетного разряда ССП (разряд 7).

4.3.4.4. Двухадресные команды

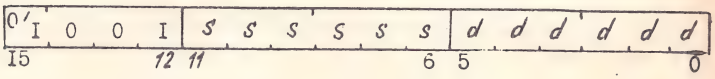
Использование двухадресных команд обеспечивает экономию машинного времени и сокращает количество команд в программе. Список двухадресных команд содержит 4 арифметические команды и 4 логические команды.

Арифметические команды.

MOV — 01SSDD

MOVB — 11SSDD

Пересылка



Действие: (dst) ← (src)

Признаки:
- N – устанавливается, если (src)<0, в противном случае очищается;
- Z – устанавливается, если (src) = 0, в противном случае очищается,
- V – очищается,
- C – не изменяется.

Описание: Операнд источника (src) пересылается по адресу операнда приемника. Прежнее содержимое ячейки dst теряется. Содержимое ячейки src не изменяется. При операциях с байтами команда MOVB с использованием регистрового метода адресации (единственная среди байтовых команд) расширяет старший разряд младшего байта (расширение знака). Все разряды старшего байта устанавливаются или сбрасываются в зависимости от

того, установлен или сброшен старший (знаковый) разряд младшего байта. В других случаях MOVB оперирует с байтами так, как MOV со словами.

Примеры: MOVX, RI загружает RI содержимым ячейки памяти X.

MOV#20, R0 загружает число 20 в регистр R0.

MOV*#20, -(R6) записывает операнд, содержащийся в ячейке с адресом 20, в стек.

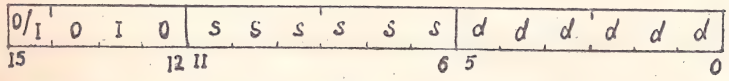
MOV (R6)+, *#I77566 выбирает операнд из стека и записывает его в ячейку I77566.

MOV R1, R3 выполняет межрегистровую передачу.

CMP 02SSDD

CMPB I2SSDD

Сравнение



Действие: (src) - (dst)

Признаки: N - устанавливается, если результат < 0, в противном случае очищается;

Z - устанавливается, если результат = 0, в противном случае очищается;

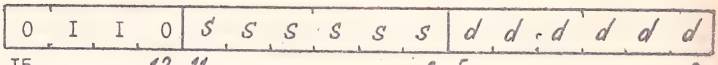
V - устанавливается, если было арифметическое переполнение. Это может произойти в случае, когда операнды были противоположного знака, а знак результата совпадает со знаком операнда приемника (dst), в противном случае очищается;

C – очищается, если был перенос из старшего разряда результата, в противном случае устанавливается.

Описание: Сравнивает операнды источника и приемника и изменят признаки, которые затем могут быть использованы для команд условных переходов. Оба операнда не изменяются. Единственным действием является изменение признаков. За командой сравнения обычно следует команда условного ветвления. Заметим, что в отличие от команды вычитания, порядок действия следующий: (src) – (dst), а не (dst) – (src).

ADD 06SSDD

Сложение



Действие: (dst) ← (src) + (dst)

Признаки: N – устанавливается, если результат < 0, в противном случае очищается;

Z – устанавливается, если результат = 0, в противном случае очищается;

V – устанавливается, если в результате операции произошло арифметическое переполнение. Это происходит, когда операнды были одного знака, а результат получился противоположного знака, в противном случае очищается;

C – устанавливается, если был перенос из старшего разряда результата, в противном случае очищается.

Описание: Операнд источника (*src*) складывается с операндом приемника (*dst*)и результат записывается по адресу операнда приемника.

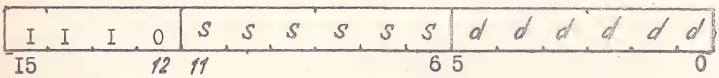
Первоначальное содержимое *dst* теряется. Содержимое *src* не изменяется. Сложение выполняется в двоичном дополнительном коде.

Пример: ADD R1, R2

До операции:	После операции:
(R1) = 000020	(R1) = 000020
(R2) = 000060	(R2) = 000100

SUB 16SSDD

Вычитание



Действие: *(dst)* ← *(dst)* - *(src)*

Признаки: *N* - устанавливается, если результат < 0, в противном случае очищается;

Z - устанавливается, если результат = 0, в противном случае очищается,

V - устанавливается, если в результате операции произошло арифметическое переполнение. Это происходит, когда операнды были противоположного знака, а результат получен одного знака с операндом источника, в противном случае очищается,

C – очищается, если был перенос из старшего разряда результата, в противном случае устанавливается.

Описание: Из операнда приемника вычитается операнд источника (*src*) и результат записывается по адресу *dst*.

Первоначальное содержимое *dst* теряется, а содержимое *src* остается без изменения. При арифметических операциях с удвоенной точностью, установка С-разряда означает заем единицы из старшей части вычитаемого.

Пример: *SUB* RI, *R2*

До операции:

(RI) = OIIIII

(R2) = OI2345

NZVC
IIII

После операции:

(RI) = OIIIII

(R2) = OOI234

NZVC
0000

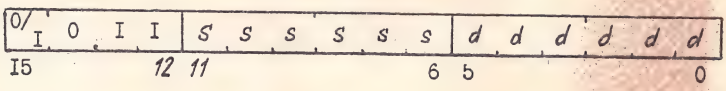
Логические команды

Из четырех логических команд три имеют такой же формат, как и двухадресные арифметические команды. Четвертая команда имеет специфический формат. Логические команды позволяют осуществлять поразрядную обработку данных.

BIT 03*SSDD*

BITB I3*SSDD*

Проверка разрядов



3.858.382 ТО

Действие: $(src) \wedge (dst)$

Признаки: N – устанавливается, если старший разряд результата установлен, в противном случае очищается;

Z – устанавливается, если все разряды результата = 0, в противном случае очищается;

V – очищается;

C – не изменяется.

Описание: Выполняет логическую функцию "И" над (src) и (dst), изменяя соответствующим образом признаки. Оба операнда не изменяют своего значения.

Команда BIT используется для проверки состояния разрядов операнда (src), для которых установлены соответствующие разряды в операнде (dst).

Пример: BIT 30, R3

До операции:	После операции:
(R3) = 0000000000011000	(R3) = 0000000000011000
NZVC	NZVC
1111	0001

BIC 04SSDD

BICB 14SSDD

Очистка разрядов

3.858.382 ТО

Действие: $(dst) \leftarrow (\overline{src}) \wedge (dst)$

Признаки: N - устанавливается, если старший разряд результата установлен, в противном случае очищается;

Z - устанавливается, если все разряды результата очищены, в противном случае очищается;

V - очищается;

C - не изменяется.

Описание: Очищает каждый разряд операнда (*dst*), соответсвующий установленному разряду операнда (*src*). Первоначальное содержимое *dst* теряется. Содержимое *src* не изменяется.

Пример: BIC, R3, R4

До операции:

(R3) = 001234

(R4) = 001111

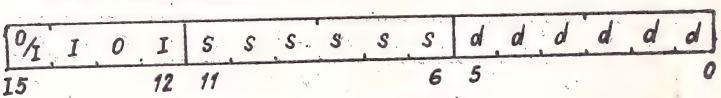
После операции:

(R3) = 001234

(R4) = 000101

	NZVC	NZVC
	1111	0001
BIS		05SSDD
BISB		15SSDD

Логическое сложение



Действие: $(dst) \leftarrow (src) \vee (dst)$

Признаки N - устанавливается, если старший разряд результата установлен, в противном случае

очищается;

Z – устанавливается, если все разряды результата очищены, в противном случае очищается,

V – очищается,

C – не изменяется.

Описание: Выполняет логическую операцию "ИЛИ" над содержимым *src* и *dst* и записывает результат по адресу *dst* . Разряды *(dst)* устанавливаются в "I", если соответствующие им разряды *(src)* находятся в "I". Прежнее содержимое *dst* теряется, а содержимое *src* остается неизменным.

Пример: *BIS RC, RI*

До операции:

(R0) = 001234

(RI) = 001111

NZVC
0000

После операции:

(R0) = 001234

(RI) = 001335

NZVC
0000

До операции:

(R0) = 0000001010011100

(RI) = 0000001001001001

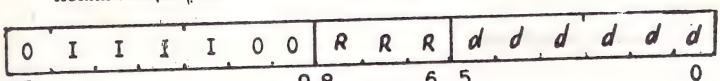
После операции.

RI) = 0000001011011101

074RDD

XOR

Исключающее ИЛИ



Действие: *(dst)* ← *R* ⊻ *(dst)*

Признаки: N – устанавливается, если результат < 0, в противном случае очищается;

Z - устанавливается, если результат = 0, в противном случае очищается;

V - очищается;

C - не изменяется.

Описание: Выполняет операцию "исключающее ИЛИ" над содержимым указанного регистра и содержимым *dst*. Результат записывается в *dst*. Содержимое регистра *R* не изменяется.

Пример: XOR R0, R2

До операции	После операции.
(R0) = 100I234	(R0) = 00I234
(R2) = 00IIII	(R2) = 000325
NZVC IIII	NZVC 000I

До операции: (Ro) = 0000001010011100

(R2) = 0000001001001001

После операции. (R2) = 0000000011010101

4.3.4.5. Команды управления программой.

К командам управления программой относятся команды ветвлений, перехода к программе, возврата из подпрограммы, безусловного перехода и др.

Команды ветвления.

Эти команды вызывают ветвление по адресу, являющемуся суммой смещения (умноженного на два) и текущего содержимого СК, если условие ветвления выполняется.

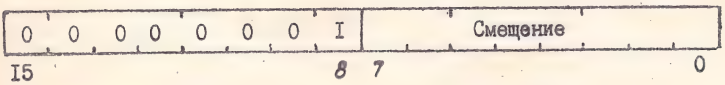
Смещение показывает, на сколько ячеек нужно перейти относительно текущего содержимого СК в ту или другую сторону. Так как слова имеют чётные адреса, то для получения истинного испол-

нительного адреса смещение необходимо умножить на два перед прибавлением к СК, который всегда указывает на слово. Старший разряд смещения (разряд 7) является знаковым разрядом. Если он установлен, смещение отрицательное, ветвление происходит в сторону уменьшения адреса (в обратном направлении). Если в 7-ом разряде содержится 0, смещение положительное, и ветвление происходит в сторону увеличения адресов (прямом направлении).

Восьмиразрядное смещение позволяет производить ветвление в обратном направлении максимально на 200_8 слов от слова, на которое указывает текущее содержимое СК, и на 177_8 слов в прямом направлении.

BR 000400 + XXX

Ветвление безусловное



Действие: (СК) ← (СК) + 2XXX

Признаки: не изменяются.

Описание: Обеспечивает способ передачи управления в программе ячейки, адрес которой находится в ограниченной области, с помощью одного слова команды.

Новое содержимое СК = текущее содержимое СК + 2 х(смещение), где текущее содержимое СК = адрес команды ветвления + 2.

Пример:

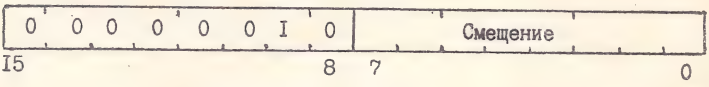
Адрес	Код команды	Смещение
500	000402	002
502		

В данном примере управление передается ячейке с адресом СК = 502 + 2 x 2 = 506.

Простые условные ветвления.

BNE 001000 + XXX

Ветвление, если не равно (нулю)



Действие: (СК) ← (СК) + 2XXX, если Z = 0

Признаки: не изменяются.

Описание: Проверяет состояние разряда Z и вызывает ветвление, если он очищен. BNE обратна по действию BEQ . Вместе с комадой CMP она используется для проверки неравенства двух величин. Вместе с командой BIT она используется для проверки того, что установленные разряды операнда источника соответствуют установленным разрядам операнда приемника. В общем случае она используется для проверки неравенства нулю результата предыдущей операции.

Пример: CMP A, B сравниваются A и B

BNE C переход, если A не равно B

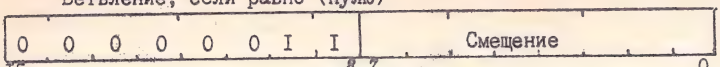
Эта последовательность команд будет давать переход на C, если A ≠ B, а последовательность:

ADD A,B, складываются A и B

BNE C переход, если A + B ≠ 0 – будет давать переход, если A + B ≠ 0.

BEQ 001400 + XXX

Ветвление, если равно (нулю)



Действие: $(СК) \leftarrow (СК) + 2XXX$, если $Z = 1$

Признаки: Не изменяются.

Описание: Проверяет состояние разряда Z и вызывает ветвление, если он установлен. Вместе с командой CMP она используется для проверки равенства двух величин. Вместе с командой BIT используется для проверки того, что очищенные разряды операнда источника соответсвуют установленным разрядам операнда приемника. В общем случае эта команда используется для проверки равенства нулю результата предыдущей операции.

Пример: CMP A, B сравнивает A и B

BEQ C переход, если они равны

Если A - B = 0, то управление передается ячейке C.

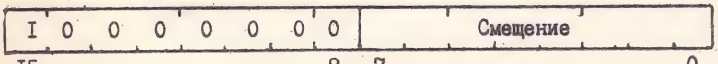
ADD A, B складываются A и B

BEQ C переход, если A + B = 0.

BPL Управление передается ячейке C, если A + B = 0

I00000 + XXX

Ветвление, если плюс



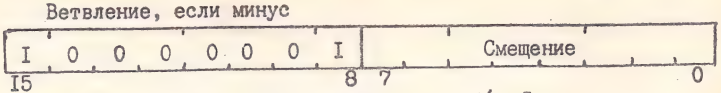
Действие: $(СК) \leftarrow (СК) + 2XXX$, если $N = 0$

Признаки: не изменяются.

Описание: Проверяет разряд N и вызывает ветвление, если он очищен. BPL обратна по действию команде BMI.

BMI 100400+XXX

Ветвление, если минус



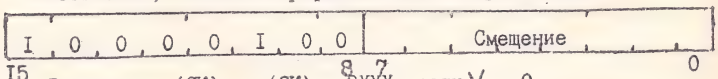
Действие: (СК) ← (СК) + 2XXX, если N = I

Признаки: не изменяются.

Описание: Проверяет состояние N разряда и вызывает ветвление, если он установлен. Она используется для проверки знака (старший разряд) результата предыдущей операции.

BVC I02000 + XXX

Ветвление, если нет арифметического переполнения



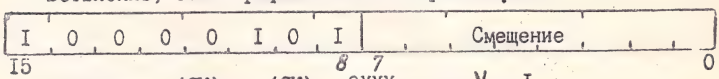
Действие: (СК) ← (СК) + 2XXX, если V = 0

Признаки: не изменяются.

Описание: Проверяет состояние разряда V и вызывает ветвление, если он очищен.

BVS I02400 + XXX

Ветвление, если арифметическое переполнение



Действие: (СК) ← (СК) + 2XXX, если V = I

Признаки: Не изменяются.

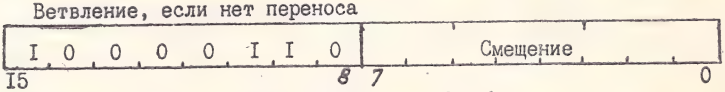
Описание: Проверяет состояние разряда V и вызывает ветвление, если он установлен.

BVS используется для обнаружения арифметического переполнения в результате исполнения предыдущей операции.

BVS обратна по действию команде BVC.

BCC I03000 + XXX

Ветвление, если нет переноса



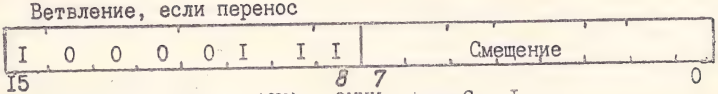
Действие: (СК) ← (СК) + 2XXX, если C = 0

Признаки: Не изменяются.

Описание: Проверяет состояние разряда C и вызывает ветвление, если он очищен.

BCS 103400 + XXX

Ветвление, если перенос



Действие: (СК) ← (СК) + 2XXX, если C = I

Признаки: Не изменяются

Описание: Проверяет разряд C и вызывает ветвление, если он установлен. BCS используется для проверки наличия переноса в результате предыдущей операции.

BCS обратна по действию команде BCC.

Условные ветвления по результату операций над числами. Особые комбинации разрядов признаков проверяются с помощью команд условного ветвления по результату операций над числами. Эти команды используются для проверки результатов команд, в которых операнды рассматриваются как двоичные числа, имеющие знак.

Заметим, что сравнение чисел, имеющих знак, отличаются от сравнения чисел без знака тем, что в арифметике, использующей дополнительные коды, последовательность чисел следующая:

наибольшее	077777
	077776
	.....
	.....
положительные	000001
	000000
отрицательные	177777
	.....
	.....
	100001
наименьшее	100000

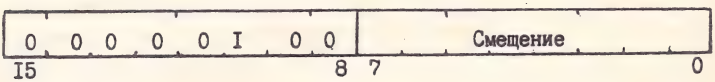
тогда как для I6-разрядных чисел, не имеющих знака, последовательность следующая:

наибольшее	I77777
	.....
	.....
	000002
	000001
наименьшее	000000

Командами условного ветвления по результату операции над числами являются следующие: BGE, BLT, BGT, BLE

BGE 002000 + XXX

Ветвление, если больше или равно (нулю)



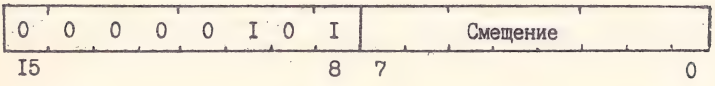
Действие: (СК) ← (СК) + 2XXX, если $N \forall V = 0$

Признаки: не изменяются.

Описание: Вызывает ветвление, если оба разряда признаков N и V установлены или очищены. Таким образом, BGE всегда будет вызывать ветвление, если она следует за операцией сложения двух положительных чисел. BGE будет также вызывать ветвление по нулевому результату.

BLT + XXX

Ветвление, если меньше (нуля)



Действие: (СК) ← (СК) + 2XXX, если $N \neq V$ =I

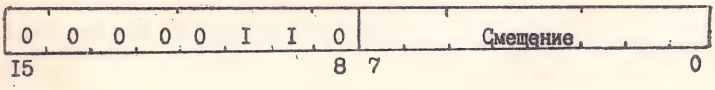
Признаки: не изменяются.

Описание: Вызывает ветвление, если результат операции "исключающее ИЛИ" над содержимым разрядов N и V равен I. Команда BLT обратна по действию команде BGE.

Таким образом, BLT будет всегда вызывать ветвление, если она следует за операцией сложения двух отрицательных чисел, даже если происходит переполнение. В частности, BLT будет всегда вызывать ветвление, если она следует за командой сравнения отрицательного операнда источника и положительного операнда назначения, даже если произошло переполнение. BLT никогда не будет вызывать ветвления, если она следует за командой сравнения (CMP) положительного операнда источника и отрицательного операнда назначения. BLT не будет вызывать ветвления, если результат предыдущей операции равен 0 без переполнения.

BGT 003000 + XXX

Ветвление, если больше (нуля)



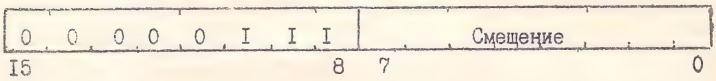
Действие: (СК) ← (СК) + 2XXX, если $Z \vee (N \vee V) = 0$

Признаки: не изменяются.

Описание: Команда BGT подобна команде BGE, за исключением того, что BGT не будет вызывать ветвления по нулевому результату.

BLE 003400 + XXX

Ветвление, если меньше или равно (нулю)



Действие: (СК) ← (СК) + 2XXX, если $Z \vee (N + V) = I$

Признаки: не изменяются.

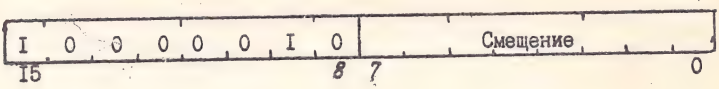
Описание: Команда BLE подобна команде BLT, но дополнительно будет вызывать ветвление, если результат предыдущей операции был равен нулю.

Условные ветвления по результату операции над кодами.

Условные ветвления по результату операций над кодами обеспечивают методы проверки результата операций сравнения операндов, рассматриваемых как величины без знака.

BHI 101000 + XXX

Ветвление. если больше



Действие: (СК) ← (СК) + 2XXX, если C=0 и $\bar{Z}$ =0

Признаки: не изменяются.

Описание: Вызывает ветвление, если предыдущая операция не вызвала переноса и появления нулевого результата. Это происходит при операциях сравнения (CMP), когда операнд источника больше операнда назначения.

BLOS 101400 + XXX

Ветвление, если меньше или равно

Действие: (СК) ← (СК) + 2XXX, если C∨$\bar{Z}$ =I

Признаки: не изменяются.

Описание: Вызывает ветвление, если предыдущая операция вызывает перенос или появление нулевого результата. Команда BLOS является обратной по действию команде BHI.

Ветвление будет происходить, если операнд источника меньше или равен операнду назначения.

BHIS 103000 + XXX

Ветвление, если больше или равно.

По своему действию команда BHI идентична команде BCC. Эта мнемоника вводится только для удобства.

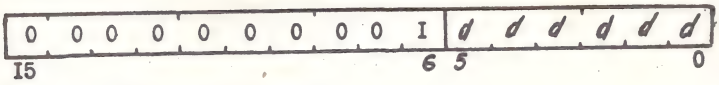
BLO 103400 + XXX

Ветвление, если меньше

По своему действию команда BLO идентична команде BCS. Эта мнемоника вводится только для удобства.

JMP 0001DD

Безусловный переход



Действие: (ск) ← (dst)

Признаки: не изменяются.

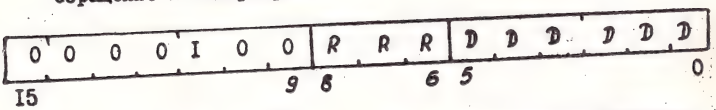
Описание: Команда JMP обеспечивает возможность перехода программы на любую ячейку памяти с использованием всех методов адресации, за исключением регистрового. Использование регистровой адресации вызывает прерывание программы по условию "запрещенная команда" через адрес вектора 4. Метод косвенной адресации может применяться и вызывает передачу управления программой по адресу, содержащемуся в указанном регистре. Заметим, что команды - это полные слова и поэтому должны выбираться из ячеек с четным адресом. Косвенно-индексный метод адресации позволяет командой JMP передать управление по адресу, являющемуся элементом таблицы адресов.

Команды обращения к подпрограмме и выхода из подпрограммы.

Эти команды обеспечивают возможность автоматического вложения программ, выход из подпрограмм, многократный вход в подпрограмму. Подпрограммы могут обращаться к другим подпрограммам (или к самим себе) без специального хранения адресов возврата. Процедура обращения к подпрограмме и выхода из нее не изменяет подпрограмму. Это позволяет использовать одну и ту же подпрограмму несколькими процессами, осуществляющими прерывание программы.

JSR 004RDD

Обращение к подпрограмме



Действие: (УС) ← (R); запись содержимого указанного регистра в стек;

(R) ← (СК), счетчик команд (СК) содержит адрес ячейки, следующей за командой JSR, этот адрес заносится в регистр R.

(СК) ← (dst); занесение в счетчик команд нового содержимого, определяющего начальный адрес подпрограммы

Признаки: не изменяются

Описание: При выполнении команды JSR старое содержимое указанного регистра ("указателя связи") автоматически засылается в стек, и новая связующая информация поступает в регистр. Таким образом, обращение к подпрограммам, вложенным в подпрограммы на любую глубину, осуществляются с помощью регистра "указателя связи". Нет необходимости в том, чтобы задавать максимальную глубину обращения к данной подпрограмме или включать команды запоминания и восстановления "указателя связи" в каждую подпрограмму.

Так как вся связующая информация сохраняется в стеке, выполнение подпрограммы может быть прервано и подпрограмма обслуживания прерывания может обращаться к той же самой прерванной

подпрограмме. Выполнение подпрограммы может быть затем возобновлено по окончании обслуживания прерывания. Этот процесс, называемый вложением, может продолжаться до любого уровня.

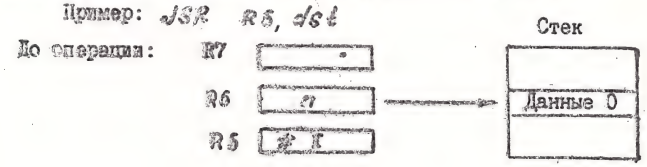
Обращение к подпрограмме по команде JSR может осуществляться с помощью автоинкрементной адресации (если каждый последующий вход в подпрограмму осуществляется через ячейку, адрес которой на 2 больше предыдущего) или индексной адресации (если вход в подпрограмму осуществляется по адресам, расположенным в произвольном порядке). Оба эти метода могут быть также косвенными.

Команда *JSR, СК, #sl* является особым случаем обращения к подпрограмме. В этом случае не изменяется содержимое ни одного из общих регистров, кроме СК

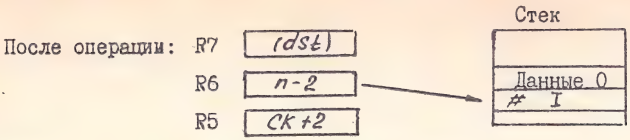
Другим, особым случаем команды JSR является JSR СК, *(SP)+, при выполнении которой последняя заполненная ячейка стека и СК обмениваются содержимым. Использование этой команды позволяет двум подпрограммам попеременно передавать управление друг другу и каждый раз возобновлять работу с того места, где осуществлялась передача управления. Такие подпрограммы называются "со-программами".

Возврат из подпрограммы осуществляется командой RTS. По команде RTS содержимое регистра передается в СК, а содержимое верхней ячейки стека – в указанный регистр.

Пример: *JSR R5, #sl*

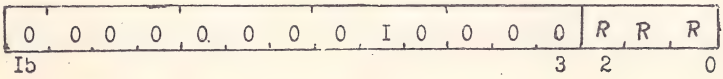


3.858.382 ТО



RTS 00020R

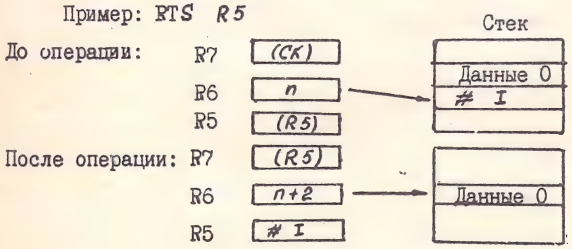
Возврат из подпрограммы



Действие: (СК)←(R)
(R)←(УС)↑

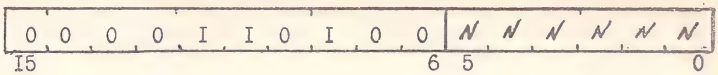
Описание: Загружает содержимое регистра (R) в СК, после чего извлекает верхний элемент стека и засылает его в указанный регистр. Возврат из подпрограммы обычно выполняется через тот же самый регистр, который используется при обращении к ней. Таким образом, выход из подпрограммы, обращение к которой осуществлялось командой JSR СК, dst выполняется командой RTS СК, а выход из подпрограммы, обращение к которой осуществлялось командой JSR R5, dst с использованием любого из методов адресации, выполняется командой RTS R5.

Пример: RTS R5



MARK 0064NN

Восстановление УС



Действие: (УС) ← (СК) + 2×NN

(СК) ← (R5)

(R5) ← (УС)↑

Признаки: не изменяются.

Описание: Эта команда используется для облегчения выхода из подпрограммы.

При использовании стека для записи в него параметров команда MARK восстанавливает указатель стека (УС) во время выхода из подпрограммы.

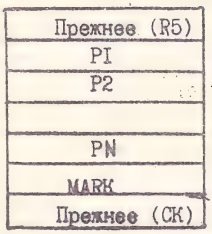
Пример:
```
MOV  R5, -(УС);   предыдущее (R5) заносится в стек
MOV  P1, -(УС),   для использования
MOV  P2, -(УС);   в программе в стек заносится N
                  параметров
MOV  PN, -(УС);
MOV  MARK, -(УС); команда MARK заносится в стек
MOV  УС, R5       в R5 запоминается адрес команды
                  MARK
JSR  СК, dst      обращение к подпрограмме,
                  выполнение подпрограммы с начальным
                  адресом, определяемым (dst)
```

3.858.382 ТО

RTS R5 возврат из подпрограммы

После выполнения команды JSR CK, dst

стек будет выглядеть следующим образом:

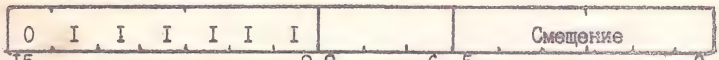


RTS

Команда загрузит счетчик команд содержимым регистра R5 (адрес команды MARK) и, выбрав из стека прежнее содержимое СК, загрузит в R5. После чего команда MARK, восстановив указатель стека, загрузит СК новым содержимым R5 (в R5 в это время находится прежнее содержимое СК). Под восстановлением УС подразумевается здесь загрузка в него нового содержимого, которое указывало бы на последнюю заполненную ячейку стека перед тем, как возникла необходимость записи в стек N констант.

SOB 077RNN

Вычитание единицы и ветвление



Действие: (R)←(R) - I, если результат ≠ 0, (СК)←(СК)-2NN

если результат = 0, (СК)←(СК)

Признаки: не изменяются.

Описание: Содержимое регистра уменьшается на единицу.

Если результат ≠ 0, в счетчик команд загружается новое содержимое, определяемое вычитанием из текущего содержимого СК удвоенного смещения.

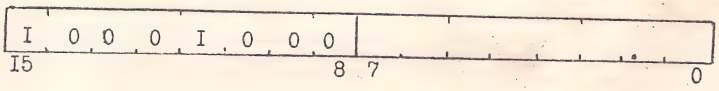
В команде SOB смещением является шестиразрядное положительное число. Эта команда может быть эффективно использована для организации различного рода счетчиков. Следует заметить, что команда SOB может быть использована для передачи управления в прямом направлении.

4.3.4.6. Команды прерывания программы.

Команды прерывания обеспечивают обращения к моделирующим программам, программам управления вводом-выводом, программам отладки и программам, разработанным пользователем. Когда происходит прерывание, текущее содержимое счетчика команд и содержимое регистра состояния процессора записываются в стек. Новое содержимое СК и РСП загружается из вектора прерывания, состоящего из двух слов. При выходе из прерывания используются команды RTI и RTT, которые восстанавливают СК и РСП, извлекая их прежнее содержимое из стека. Векторы прерывания расположены по фиксированным, приписанным каждому виду прерывания, адресам..

EMT 104000- 104377

Командное прерывание для системных программ



Действие: ↓(УС) → (РСП)
↓(УС) → (СК)
(СК) → (30)
(РСП) → (32)

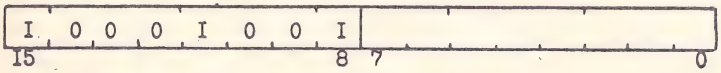
Признаки: загружаются из вектора прерывания.

Описание: Команды с кодами операций от 104000 до 104377 являются командами EMT и могут быть использованы для передачи информации в моделирующую программу (т.е. информации о функции, которая должна быть выполнена)

Вектор прерывания для EMT находится по адресу 30. Новое содержимое СК берется из ячейки с адресом 30, а новое содержимое РСП - из ячейки с адресом 32.

TRAP 104400 - 104777

Командное прерывание



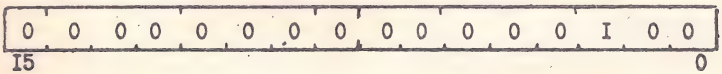
Действие: ↓(УС) ← (РСП)
↓(УС) ← (СК)
(СК) ← (34)
(РСП) ← (36)

Признаки: Загружаются из вектора прерывания.

Описание: Команды с кодами операций от 104400 до 104777 являются командами TRAP, которые по своему действию идентичны командам EMT, за исключением того, что вектор прерывания команды TRAP имеет адрес 34.

IOT 000004

Командное прерывание для ввода-вывода



Действие: ↓(УС) ← (РСП)

↓(УС) ← (СК)

(СК) ← (20)

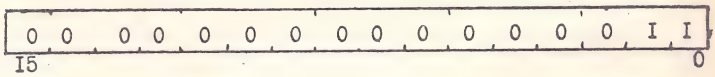
(РСП) ← (22)

Признаки: загружаются из вектора прерывания.

Описание: осуществляет прерывание с вектором прерывания, расположенным по адресу 20. Используется для обращения к подпрограмме управления вводом-выводом.

ВРТ 000003

Командное прерывание для отладки



Действие: ↓(УС) ← (РСП)

↓(УС) ← (СК)

(СК) ← (I4)

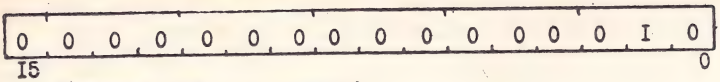
(РСП) ← (I6)

Признаки: загружаются из вектора прерывания.

Описание: выполняется прерывание с вектором, расположенным по адресу I4. Используется для обращения к подпрограммам отладки. Пользователю запрещается употребление кода 000003 в программах, которые выполняются под управлением подпрограмм отладки.

RTI 000002

Возврат из прерывания



Действие: (СК) ← (УС)↑

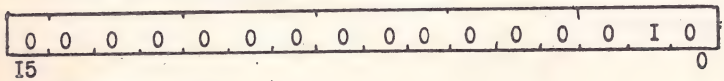
(РСП) ← (УС)↑

Признаки: загружаются из стека.

Описание: используется для выхода из подпрограмм обслуживания внешних и внутренних прерываний. Содержимое СК и РСП восстанавливается с помощью стека. Если при выполнении этой команды будет установлен Т-разряд РСП, то следующая после RTI команда выполняться не будет.

RTT 000006

Возврат из прерывания



Действие: (СК) ← (УС)↑

(РСП) ← (УС)↑

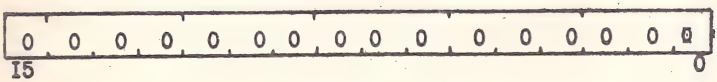
Признаки: загружаются из стека.

Описание: эта команда по своему действию идентична команде RTI за исключением того, что при установке Т-разряда РСП прерывание будет иметь место после того, как выполнится первая команда, следующая за RTT.

4.3.4.7. Команды управления машиной.

HALT 000000

Останов

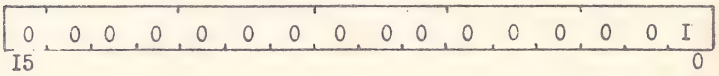


Признаки: не изменяются.

Описание: центральный процессор переходит в режим опроса пультового терминала. В счетчике команд находится адрес следующей команды, которая должна быть исполнена. Машина будет находиться в состоянии ОСТАНОВ до тех пор, пока оператор не продолжит выполнение программы при помощи команды пультового терминала.

WAIT 000001

Ожидание



Признаки: не изменяются.

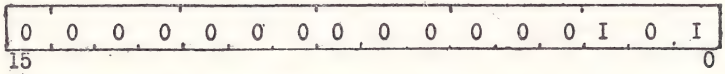
Описание: Применение этой команды обеспечивает наиболее быструю передачу канала внешнему устройству при поступлении запроса от него. По команде WAIT процессору запрещается выбирать команды из памяти. Это позволяет наиболее быстро осуществить обмен между внешними устройствами и памятью, т к. процессор не вносит задержки при обслуживании запроса канала на время, когда он освобождает занятый им канал. При выполнении команды WAIT, как и при выполнении всех других команд, в СК содержится адрес команды, следующей за командой WAIT. Таким образом, когда по прерыванию вызывается передача содержимого СК и РСП в стек, адрес команды, следующей за командой WAIT, сохраняется.

Выход из подпрограммы, обслуживающей прерывание, (т.е. выполнение команды RTI или RTT), вызовет возобновление

прерванного процесса с команды, следующей за командой WAIT.

RESET 000005

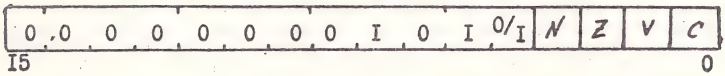
Сброс внешних устройств



Признаки: не изменяются.

Описание: по этой команде вырабатывается 10 мкс импульс СБРОС, который передается в канал. Все внешние устройства, присоединенные к каналу, устанавливаются в состояние, которое они имеют после включения питания. Через 90 мкс после импульса СБРОС процессор возобновляет свою работу.

4.3.4.8. Команды изменения признаков



Описание: выше приведен формат команд изменения признаков. Разряды признаков, соответствующие установленным разрядам в команде изменения признаков (разряды 0-3), изменяются в соответствии с состоянием разряда 4 (разряда установки/сброса). Эти разряды РСП устанавливаются, если установлен 4-й разряд. Если же он очищен, то очищаются.

Ниже перечисляются команды изменения признаков.

Обозначение	Операция	Код
CLN	Очистка N	000250
CLZ	Очистка Z	000244
CLV	Очистка V	000242

Обозначение	Операция	Код
CLC	Очистка C	000241
CCC	Очистка всех разрядов (N,Z,V,C)	000257
SEN	установка N	000270
SEZ	установка Z	000264
SEV	установка V	000262
SEC	Установка C	000261
SCC	Установка всех разрядов (N,Z,V,C)	000277
NOP	нет операции	000240

Комбинации указанных выше операций очистки или установки, соединенных по схеме "ИЛИ", могут образовывать комбинированные команды.

4.3.5. Внутренние прерывания программы.

а) Прерывание по резервной команде – это прерывание, вызывающееся при попытке выполнить команды, коды которых в данной ЭВМ не используются и зарезервированы для дальнейших расширений системы команд (резервные команды). При попытке выполнения команды с неправильно заданными методами адресации (запрещенной команды) также будет вызываться прерывание.

Запрещенными командами, являются, например, команды jMP и jSP с регистровым методом адресации, они будут вызывать прерывание программы по вектору с адресом 4. Прерывание по резервной команде осуществляется по вектору с адресом 10.

б) Прерывание по ошибке обращения к каналу. Это прерывание возникает при попытке обращения по адресу несуществующих ячеек памяти или регистра внешнего устройства. Прерывание по ошибке обращения к каналу будет иметь место в том случае, если

пассивное (управляемое) устройство не способно передать или принять информацию за отведенное ему на это время. Это прерывание использует вектор с адресом 4.

в) Прерывание по Т-разряду вызывается при установке 4-го разряда ССП. Обычно это прерывание с адресом вектора I4 используется для отладки программы. Команда, которая должна выполняться следом за установившей Т-разряд командой RTT, будет выполняться до конца, а затем произойдет внутреннее прерывание программы с вектором по адресу I4. Далее описываются особые случаи прерывания по Т-разряду в зависимости от типа команды, выполняемой после установки Т-разряда (прослеживаемой команды):

- прослеживаемая команда очищает Т-разряд;
- прослеживаемая команда устанавливает Т-разряд;
- прослеживаемая команда является командой прерывания;
- при выполнении прослеживаемой программы вызывается прерывание по ошибке обращения к каналу;
- в промежутке времени между установкой Т-разряда и началом выполнения прослеживаемой команды вызывается внешнее прерывание программы;
- прослеживаемой командой является команда HALT;
- прослеживаемой командой является команда WAIT;
- прослеживаемой командой является команда возврата из прерывания.

Команда очищает Т-разряд.

Перед выборкой этой команды Т-разряд РСП был установлен. По окончании выполнения этой команды все же произойдет прерывание программы, но запоминаемое в стеке слово состояния процессора будет иметь очищенный Т-разряд.

Команда устанавливает Т-разряд.

Поскольку Т-разряд был установлен, дополнительная его установка не приводит ни к каким изменениям. После выполнения команды происходит прерывание программы.

Прослеживаемой командой является команда прерывания.

Происходит прерывание программы с соответствующим вектором и полностью выполняется программа обслуживания этого прерывания. Если выполнение этой программы заканчивается с помощью команды RTI или другим путем, при котором происходит восстановление запомненного в стеке слова состояния программы, может вновь быть установлен Т-разряд РСП. В этом случае выполняется следующая за прослеживаемой команда и, если не встречается описанных выше случаев, происходит прерывание по Т-разряду.

При выполнении команды происходит ошибка обращения к каналу.

Выполняются действия, аналогичные тем, что происходят при попытке выполнения программы прерывания. Единственным отличием является то, что при обслуживании прерывания по ошибке обращения к каналу выход из программы обслуживания с помощью команды RTI не происходит, поэтому прерывание по Т-разряду может не вызываться.

Если в промежутке между выборкой нового содержимого СК и РСП из вектора и первой командой возникает требование прерывания, оно будет удовлетворено немедленно. Для предотвращения прерывания по запросу внешнего устройства разряд 7 РСП

3,858.382 ТО

должен быть установлен в "I".

Следует избегать выполнения команды WAIT , когда установлен Т-разряд, т.к. процессор переходит в состояние, из которого его можно вывести только непродолжительным низким уровнем сигнала К. ПОСТН В.

При исполнении команды HALT происходит останов машины. В счетчике команд при этом находится адрес следующей команды, которая должна быть исполнена. Если команда HALT является прослеживаемой командой, прерывание произойдет сразу после возобновления работы.

Если прослеживаемой командой является команда RTT, прерывание не наступит до окончания выполнения следующей за RTT команды.

г) Прерывание по нарушению питания

Это прерывание может возникнуть при поступлении в процессор, находящийся в режиме программной работы, сигнала нарушения питания. Прерывание по нарушению питания использует вектор с адресом 24.

В случае одновременного возникновения различных условий внутренних и внешних прерываний в ЭВМ принят следующий порядок их обслуживания:

- прерывание по ошибке обращения к каналу;
- регенерация ОЗУ;
- командное прерывание;
- прерывание по Т-разряду;
- аппаратный останов;
- прерывание по нарушению питания;

- прерывание по таймеру;
- прерывание от внешнего устройства.

4.3.6. Организация стека.

Стек является динамичным последовательным списком данных, помещенным в специально отведенную для него область оперативной памяти, доступ к которому осуществляется только с одной стороны. В основу организации стека положен принцип: "записанный последним читается первым". В ЭВМ "Электроника 60М" стек используется при прерываниях программы для запоминания текущего содержимого счетчика команд и слова состояния процессора, а также для временного хранения данных, часто используемых программой.

Обращение к стеку осуществляется через указатель стека (УС). Указатель стека содержит адрес последней занятой ячейки стека. Для аппаратно-реализованного стека, используемого при обслуживании прерываний программы и обращении к подпрограммам, в качестве указателя стека используется R6. При программно реализуемом стеке в качестве указателя стека может быть использован любой из регистров общего назначения. В этом случае запись в стек и выборка из стека осуществляется с помощью автодекрементного и автоинкрементного методов адресации. Ниже приводятся примеры программного использования стека для хранения информации или для выборки ее из стека.

Пример I.

```
MOV   R3,   - (УС)
MOV   R2,   - (УС)
MOV   RI,   - (УС)
MOV   R0,   - (УС)
```

С помощью этой программы реализуется задача сохранения содержимого регистров R3, R2, R1, R0 в стеке. Причем, в самом низу стека будет записано содержимое R3, затем R2, R1 и R0. Содержимое R0 находится вверху стека. В данном примере запись в стек осуществляется с помощью автодекрементного способа адресации.

Пример 2.

```
MOV  (УС)+    R0
MOV  (УС)+    R1
MOV  (УС)+    R2
MOV  (УС)+    R3
```

В этом примере восстанавливается содержимое регистров по данным, хранимым в стеке. Выборка из стека осуществляется с помощью автоинкрементного метода адресации в обратном порядке по сравнению с их записью в стек.

К стеку можно обращаться тремя способами:

1) программно: с помощью команд пересылок (см. примеры № 1 и № 2);

2) используя команды работы с подпрограммами JSR и RTS;

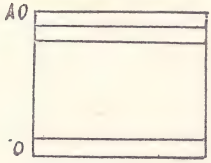
3) автоматически, при прерывании программы.

При выполнении команд JSR в стек засылается содержимое регистра, называемого "указателем связи", а при выполнении команды RTS содержимое указанного регистра восстанавливается из стека.

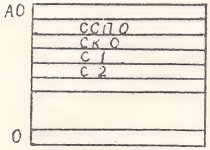
При прерывании программы стек используется для автоматического хранения текущего содержимого счетчика команд (адреса возврата на прерванную программу) и слова состояния

программы.

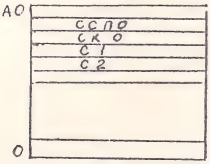
Использование стека для организации одновременного обслуживания прерывающих программу устройств иллюстрируется на следующем примере:



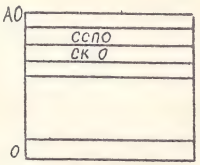
I. Выполняется программа №0. УС указывает на ячейку A0



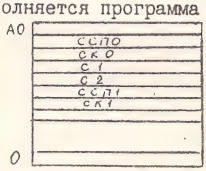
3. Программа №I использует стек для временного хранения слов C1 и C2



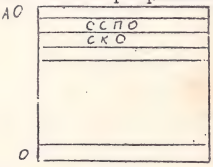
5. Программа завершена выполнением команды RTI. В СК заносится СКI, а в РСП-ССПI и возобновляется выполнение программы №I.



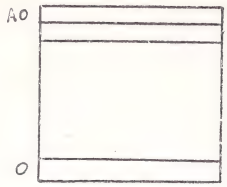
2. Программа №0, прервана при ССП=ССП0 и СК=СК0. Выполняется программа №I



4. Программа №I прервана при ССП=ССПI и СК=СКI. Выполняется программа №2.



6. Программа №I освобождает ячейки, содержащие C1 и C2



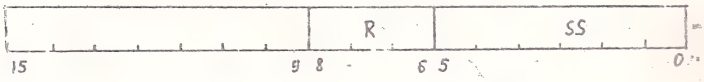
7. Программа №I завершена выполнением команды RTI, в СК заносится СКО, а в РСП-ССПО и возобновляется выполнение программы №0

4.3.7. Команды расширенной арифметики (РА) и арифметики с плавающей запятой (ПЗ)

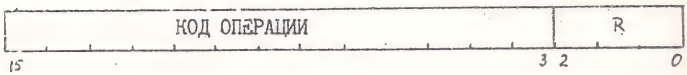
В этом разделе рассматриваются команды, позволяющие выполнять операции над 16-ти и 32-разрядными словами с фиксированной запятой и над 32-разрядными словами с плавающей запятой. Выполнение этих команд обеспечивается микропрограммами, хранящимися в третьей БИС МПЗУ.

Формат команд РА и ПЗ выглядит следующим образом:

Формат команды РА (MUL, DIV, ASH, ASHC).



Формат команды ПЗ (FADD, FSVB, FMUL, FDIV).



4.3.7.1. Команды расширенной арифметики

Для выполнения операций над числами с фиксированной запятой используются следующие команды:

MUL – умножение,

DIV – деление,

ASH – арифметический сдвиг,

ASHC – арифметический сдвиг двойного слова.

Формат чисел, используемых при выполнении команд расширенной арифметики следующий:

I6-разрядное слово

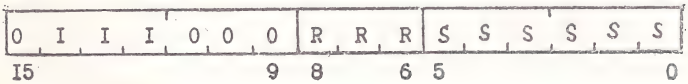
32-разрядное слово

Разряд I5 старшей части 32-разрядного слова или разряд I5 обычного I6-разрядного слова являются знаковыми разрядами.

Положительным числам соответствует 0 в знаковом разряде, отрицательным - I.

MUL 070RSS

Умножение



Действие: R, RvI R*(S2C)

Признаки: N – устанавливается, если результат < 0, в противном случае очищается,

Z – устанавливается, если результат $= 0$, в противном случае очищается,

V – очищается,

C – устанавливается, если результат меньше чем -2^{15} или больше или равен $2^{15} - 1$.

Описание: Перемножаются операнды источника и приемника, взятые в двоичнодополнительном коде. Результат помещается в регистр, используемый в качестве приемника, и в следующий за ним регистр, если регистр приемника имеет четный номер. Если же регистр приемника имеет нечетный номер, сохраняется только младшая часть результата.

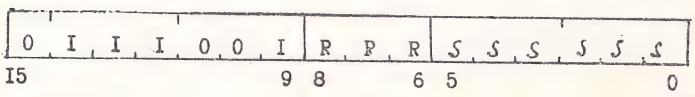
Мнемоническая запись команды умножения: MUL S,D

Пример: MUL #10, R1

До операции		После операции:	
1000) 070127	R1) 000400	1000) 070127	R1) 004000
1002) 000010		1002) 000010	

DIV 071RSS

Деление



Действие: R, R∨1 ← R, R∨1 / (S)

3.858.382 TO

Признаки: N – устанавливается, если частное < 0,
в противном случае очищается,

Z – устанавливается, если частное = 0,
в противном случае очищается,

V – устанавливается, если (src) = 0, или,
если (dst) > (src) по абсолютной величине. (В этом случае выполнение команды прекращается, т.к. частное будет превышать 15 разрядов).

C – устанавливается, если происходит попытка деления на 0.

Описание: 32-разрядное число в двоичнодополнительном коде, находящееся в регистрах R и R∨1, делится на операнд источника. Частное заносится в R, а остаток – в R∨1. После выполнения операции деления знак остатка будет таким же, как и у делимого, Следует заметить, что номер регистра R должен быть четным.

Мнемоническая запись команды деления: DIV src, R

Пример: DIV #2, R0

До операции:		После операции:	
2000)071027	R0) 000000	2000)071027	R0)010000
2002)000002	R1)020001	2002)000002	R1)000001

В R0 находится частное, а в R1 остаток.

ASH 072RSS

Арифметический сдвиг

0	1	1	1	0	1	0	R	R	R	S	S	S	S	S	S

Действие: R ← R сдвинутое на NN позиций влево или вправо, где NN – 6 младших разрядов (SZC).

Признаки: N – устанавливается, если результат < 0, в противном случае очищается,

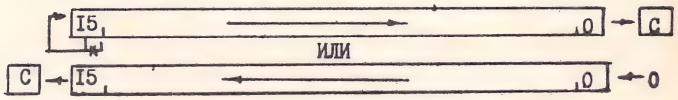
Z – устанавливается, если результат = 0, в противном случае очищается;

V – устанавливается, если после выполнения операции сдвига операнд изменил знак, в противном случае очищается,

C – загружается содержимым последнего разряда, выдвинутого из регистра.

Описание: Содержимое указанного регистра сдвигается влево или вправо на количество позиций, определяемой счетчиком сдвига. Функции счетчика сдвига выполняют 6 младших разрядов операнда источника. Его назначение может изменяться в пределах от – 32 до +31. Отрицательное значение счетчика определяет сдвиг вправо, положительное – влево.

Выполнение команды ASH иллюстрируется ниже.



Счетчик сдвига (6 младших разрядов операнда источника)

Выполняемая функция

011111	Сдвиг R на 31 позицию влево
000001	Сдвиг R на 1 позицию влево
111111	Сдвиг R на 1 позицию вправо
100000	Сдвиг R на 32 позиции вправо

Мнемоническая запись команды арифметического сдвига: ASH src,R

Пример: ASH R0,R3

До операции:

3000)072300 R0) 001234
R3)000003

После операции

3000)072300 R0)012340
R3)000003

ASHC 073RSS

Арифметический сдвиг двойного слова

Действие: R, R∨1 ← R,R∨1. Сдвинутое на NN позиций вправо или влево, где NN – 6 младших разрядов (src).

Признаки: N – устанавливается, если результат < 0, в противном случае очищается;

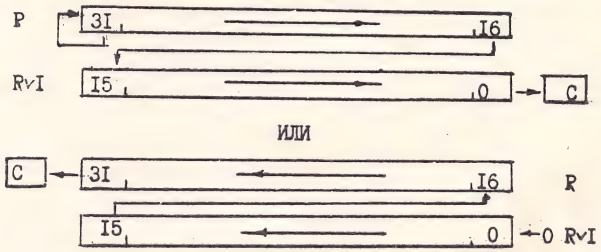
Z – устанавливается, если результат = 0, в противном случае очищается;

V – устанавливается, если после выполнения операции сдвига операнд изменил знак, в противном случае очищается;

C – загружается, содержимым последнего выдвинутого разряда 32-разрядного слова

Описание: Содержимое регистров R и R∨1 интерпретируется как одно 32-разрядное слово. Причем, младшая часть слова (разряды 00-15) содержатся в R∨1, а старшая часть (разряды 16÷31) - в R. 32-разрядное слово сдвигается вправо или влево на количесвто позиций, определяемой счетчиком сдвига. Функцию сдвига выполняют 6 младших разрядов операнда источника. Его значение может изменяться в пределах от -32 до +31. Отрицательное значение счетчика определяет сдвиг вправо, положительное - влево. Если выбранный регистр имеет нечетный номер, то R и R∨1 являются одним и тем же регистром. В этом случае сдвиг вправо будет выполняться циклически (16-разрядное слово сдвигается циклически на количество позиций, определяемое счетчиком сдвига).

Выполнение команды ASHC иллюстрируется ниже.



Мнемоническая запись команды арифметического сдвига двойного слова: ASHC src, R.

Пример: ASHC #75, R2

До операции:	После операции:
4000)073227 R2) 012345	4000)073227 R2)001234
4002)000075 R3)000000	4002)000075 R3)120000

4.3.7.2. Команды арифметики с плавающей запятой

Для выполнения операций над числами с плавающей запятой используются следующие команды:

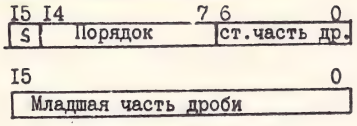
FADD – сложение с плавающей запятой;

FSUB –вычитание с плавающей запятой;

FMUL – умножение с плавающей запятой;

FDIV – деление с плавающей запятой.

Формат чисел, используемых при выполнении команд с плавающей запятой, приводится ниже



S – разряд I5 старшей части аргумента используется как знаковый разряд.

Порядок – 8 разрядов старшей части аргумента используются для задания порядка числа.

Мантисса – 23 разряда дробной части с двоичной запятой слева от самого старшего разряда дроби используется как мантисса числа. Мантисса всегда должна быть нормализована. Один, самый старший разряд справа от двоичной запятой, является скрытым (избыточным).

Двоичная запятая расположена слева от самого старшего – разряда дробной части числа (6 разряд старшей части аргумен-

та), поэтому любое значение мантиссы всегда меньше I. Нормализованная мантисса в первом после двоичной запятой разряде всегда содержит I, поэтому значения мантиссы могут изменяться в пределах от I/2 до I. Причем, этот разряд (содержащий I) представлен неявно, он лишь подразумевается. Следующий разряд мантиссы берется из 6 разряда старшей части аргумента. Результаты операции с плавающей запятой округляются в сторону увеличения абсолютной величины числа.

8-разрядное поле порядка (разряды I4 ÷ 07) старшей части аргумента) позволяет изменять порядок в пределах от - I28 до +I27. Ниже приводится соответствие десятичного, восьмеричного и двоичного кодов, соответствующих различным величинам порядка.

Десятичный код	Восьмеричный код	Двоичный код
+I27	377	II III III
+I	20I	I0 000 00I
0	200	I0 000 000
-I	I77	0I III III
-I28	000	00 000 000

Из приведенного выше примера видно, что порядок запи - сывается с избытком в 200_8 или $I28_{I0}$.

Если действительная величина порядка равна -I28, то число должно быть равно 0, вне зависимости от знакового разряда и разрядов мантиссы. В этом случае происходит аппарат-

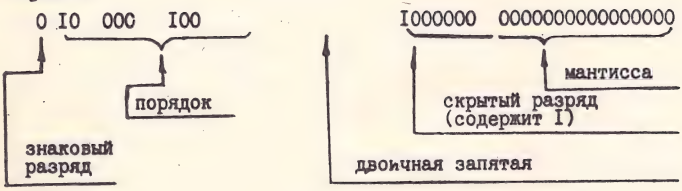
ная очистка 32-разрядного слова.

Пример числа с плавающей запятой:

$+ 12_{10} = + 1100_2 = 2^4_{10} \times 0,11_2$

$+ 12_{10} = 16 \times (1/2 + 1/4)$, следовательно

в двоичном коде $+12_{10}$ может быть представлено следующим образом:



Операнды, используемые при выполнении команд ПЗ хранятся в стеке. Содержимое регистра указывает на адреса расположения аргументов, которые расположены следующим образом:

(R) – адрес старшей части аргумента В;

(R) + 2 – адрес младшей части агрумента В;

(R) + 4 – адрес старшей части аргумента А;

(R) + 6 – адрес младшей части аргумента А.

После выполнения команды ПЗ результат помещается в стек следующим образом:

(R) + 4 – адрес старшей части результата,

(R) + 6 – адрес младшей части результата.

где (R) – первоначальное содержимое используемого РОН.

После исполнения команды ПЗ используемый РОН будет указывать на старшую часть результата, т.е. на (R) + 4.

Признаки (условные коды) устанавливаются или очищаются в соответствии с описанием выполнения команд ПЗ, которое приводится в этом подразделе. Если выполнение команды ПЗ вызывает прерывание программы, условные коды будут иметь следующие значения:

V = I, если имеет место ошибка (переполнение, антипереполнение, или попытка деления на 0),

N = I, если произошло антипереполнение или деление на 0,

C = I, если происходила попытка деления на 0,

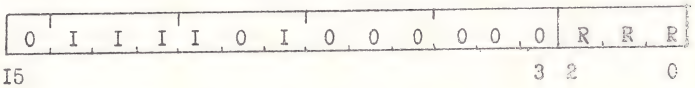
Z = 0.

	V	N	C	Z
- - - - - - - - - -	-	-	-	-
- - - - - - - - - -	-	-	-	-
Деление на 0	I	I	I	0

Прерывание, вызванное выполнением команды ПЗ, происходит через вектор с адресом 244. Если во время выполнения команды возникает требование прерывания от внешнего устройства, выполнение команды прекращается. При этом содержимое счетчика команд будет определять адрес команды ПЗ.

FADD 07500R

Сложение с плавающей запятой.



Действие: $A \leftarrow A + B$, или

$[(R) + 4, (R) + 6] \leftarrow [(R) + 4, (R) + 6] + [(R), (R) + 2]$,

если результат $\geq 2^{-I28}$, в противном случае

$[(R) + 4, (R) + 6] \leftarrow 0$.

Признаки:

N – устанавливается, если результат < 0, в противном случае очищается;

Z – устанавливается, если результат $= 0$, в противном случае очищается;

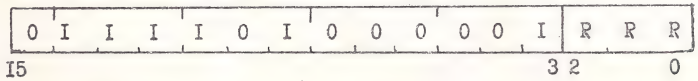
V – очищается;

C – очищается.

Описание: Складываются аргументы A и B. Результат помещается в стек на место аргумента A. В качестве указателя стека используется регистр общего назначения R.

FSUB 07501R

Вычитание с плавающей запятой



Действие: $A \leftarrow A - B$, или

$[(R) + 4, (R) + 6] \leftarrow [(R) + 4, (R) + 6] - [(R), (R) + 2]$,

если результат $\geq 2^{-I28}$, в противном случае

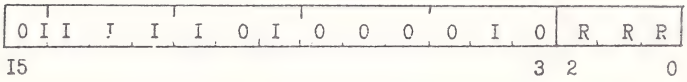
$[(R) + 4, (R) + 6] \leftarrow 0$.

Признаки: N – устанавливается, если результат < 0,
в противном случае очищается;
Z – устанавливается, если результат = 0,
в противном случае очищается;
V – очищается;
C – очищается.

Описание: Из аргумента A вычитается аргумент B. Результат помещается в стек на место аргумента A.

FMUL 07502R

Умножение с плавающей запятой

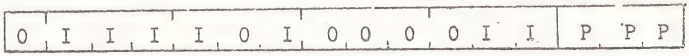


Действие: A ← A x B, или
[(R) + 4, (R) + 6] ← [(R) + 4, (R) + 6] x [(R), (R) + 2],
если результат $\geq 2^{-128}$, в противном случае
[(R) + 4, (R) + 6] ← 0.

Признаки: N – устанавливается, если результат < 0,
в противном случае очищается;
Z – устанавливается, если результат = 0,
в противном случае очищается;
V – очищается;
C – очищается.

Описание: Аргумент A умножается на аргумент B. Результат помещается в стек на место аргумента A.

Деление с плавающей запятой



Действие: A ← A/B , или

[(R) + 4, (R) + 6] ← [(R) + 4, (R) + 6] / [(R), (R) + 2]

если результат $\geqslant 2^{-128}$, в противном случае

[(R) + 4, (R) + 6] ← 0.

Признаки: N – устанавливается, если результат < 0, в противном случае очищается;

Z – устанавливается, если результат = 0, в противном случае очищается;

V – очищается;

C – очищается.

Описание: Аргумент A делится на аргумент B. Результат помещается в стек на место аргумента A. Если делитель (аргумент B) равен 0, результат в стек не записывается.

4.3.8. Время выполнения команд.

Время вычислений или выполнения программы зависит от используемых команд, применяемых методов адресации и типа выбранной памяти. Время выполнения команды Тк будет складываться из основного времени $T_Б$, времени выборки операнда источника $T_И$ и времени выборки операнда приемника $T_П$.

$$T_К = T_Б + T_И + T_П$$

Ниже приводятся таблицы, в которых указывается время выборки операндов источника и приемника в зависимости от методов адресации, основное время выполнения команд для различных методов адресации.

Времена приводятся для машин, у которых цикл равен 400 нс.

Время выборки операндов источника и приемника

Таблица 7

Метод адресации	Время выборки операнда источника, мкс		Время выборки операнда приемника, мкс	
	слово	байт	слово	байт
0	0	0	0	0
I	I,6	I,2	2,4	2,0
2	I,6	I,2	2,4	2,0
3	4,0	3,6	4,8	4,8
4	2,4	2,0	3,2	2,8
5	4,8	4,4	5,6	5,6
6	4,8	4,6	5,6	5,2
7	7,2	6,8	7,6	8,0

Примечание. Для 2 и 4 метода адресации с использованием регистров R6 и R7 при байтовых операциях прибавляется 0,4 мкс к $T_И$ и 0,8 мкс к $T_П$.

Основное время для двухадресных команд

Таблица 8

Команда	Метод адресации 0, мкс	Метод адресации I–7 мкс
MOV	4,0	2,8
ADD, SUB, XOR, BIC, BIS	4,0	4,8
CMP, BIT	4,0	3,6
MOV B	4,4	4,4
BICB, BIS B	4,4	4,4
CMPB, BITB	3,6	3,2

Основное время для одноадресных команд

Таблица 9

Команда	Метод адресации – 0, мкс	Метод адресации –I–7, мкс
CLR	4,4	4,8
INC, ADC, DEC, SBC	4,8	5,6
COM, NEG	4,8	5.2
ROL, ASR	4.4	5,2
TST	4,8	4,4
ROR	6,0	6,8

Продолжение табл. 9

Команда	Метод адресации - 0, мкс	Метод адресации I-7, мкс
ASR	6,4	7,2
CLRB, COMB, NEGB	4,4	4,8
ROLB, ASLB	4,4	4,8
INCB, DECB, SBCB, ADCB	4,4	5,2
TSTB,	4,4	4,0
ROR B	4,8	5,6
ASRB	5,2	6,8
SWAB	4,8	4,4
SXT	6,8	7,6
MFPS	5,6	7,6
MTPs	8,0	8,0*

Примечание. Для команды MTPS необходимо пользоваться временем выборки байта операнда источника и прибавить 0,4 мкс, если установлен 7 разряд.

Основное время для команд управления программой, изменения признаков и специальных команд

Таблица 10

Команда	Время выполнения команды, мкс
JMP	4,0
JSR	9,6
Все ветвления	4,0

Продолжение табл. 10

Команда,	Время выполнения команды, мкс
SOB (ветвится)	5,6
SOB (не ветвится)	4,8
Установка и очистка признаков	4,0
NOP	4,0
RTS	7,0
MARK	I3,2
RTI	I0,0*
RTT	I0,0**
TRAP, EMT	I9,2*
IOT, RPT	2I,3*
WAIT	7,2
HALT	6,4
RESET	6,8 + I2,0 + I00

Примечания.

* Прибавляется по 0,4 мкс, если в новом PC установлены 4 или 7 разряды.

** Прибавляется 2,8 мкс, если в новом PC установлен 4 разряд.

Время выборки операнда источника для команд JMP и JSR

Таблица 11

Метод адресации	Время выборки операнда источника, мкс
I	0,8

Продолжение табл. 11

Время адресации	Время выборки операнда источника, мкс
2	I,6
3	2,0
4	I,6
5	2,8
6	2,8
7	4,8

Время выборки операнда источника для
команд расширенной арифметики

Таблица 12

Метод адресации	Время выборки операнда источника, мкс
0	0,4
I	2,4
2	3,2
3	3,6
4	3,2
5	4,4
6	4,4
7	6,4

3.858.382 ТО

Основное время для команд расширенной арифметики

Таблица 13

Команда	Основное время, мкс	Примечание
MUL	от 27 до 42	Если величина сомножителей меньше 256 по абсолютной величине
	73	В наихудшем случае
DIV	89	В наихудшем случае
ASH (вправо)	II,6	+2 мкс на каждый сдвиг
ASH (влево)	I2,3	+2,8 мкс на каждый сдвиг
ASHB (вправо)	II,6	+3,2 мкс на каждый сдвиг
ASHB (влево)	II,6	+3,6 мкс на каждый сдвиг

Время выполнения команд плавающей запятой $T_К$ состоит из основного времени $T_Б$, времени сдвига двоичной запятой $T_{сд}$ и времени сдвига нормализации $T_{сн}$.

$$T_К = T_Б + T_{сд} + T_{сн}$$

Время сдвига двоичной запятой

Таблица 14

Мантисса	Время сдвига, мкс	Примечание
0 – 7	2,8	На каждый сдвиг
8 – I5	4,0	+2,8 мкс на каждый сдвиг
I6 – 23	8,0	+2,8 мкс на каждый сдвиг

Время сдвига нормализации

Таблица 15

Мантисса	Время сдвига мкс	Примечание
0- 7	2,4	На каждый сдвиг
8- I5	2,4	+2,4 мкс на каждый сдвиг
I6 - 23	4,8	+2,4 мкс на каждый сдвиг

Основное время для команд плавающей запятой

Таблица 16

Команда	Основное время, мкс	Примечание
FADD	48	
FSUB	48,5	
FMUL	от 85 до 93	Если каждый аргумент имеет точность до 7 разрядов
	I40	В наихудшем случае
FDIV	I73	Среднее время
	268	В наихудшем случае

4.3.9. Временные соотношения предоставления прямого доступа.

Максимальное время между требованием прямого доступа (ТПД) и первым циклом обмена под управлением устройства ПДП составляет 7,2 мкс, оно получается в случае команды ASR, в

которой используется I – 7 методы адресации для операнда источника.

4.3.IO. Временные соотношения предоставления прерывания

3.3.IO.I. Процессор выполняет микропрограмму регенерации памяти:

– время между требованием прерывания и предоставлением прерывания (максимум) – I35 мкс

– время между предоставлением прерывания (ПНР) и выборкой первой обслуживающей команды (максимум) – I8,8 мкс

– полное время между требованием прерывания и выборкой первой обслуживающей команды (максимум) – I53,8 мкс

4.3.IO.2. Процессор не занят выполнением микропрограммы регенерации памяти:

– время между требованием прерывания (ТПР)и предоставлением прерывания (ПНР) (максимально для команды IOT) – 2I,2 мкс

– время между предоставлением прерывания и выборкой первой обслуживающей команды (максимум) – I8,8 мкс

– полное время между требованием прерывания и выборкой первой обслуживающей команды (максимум) – 40 мкс

4.3,IO.3. Процессор не занят микропрограммой регенерации памяти, но в программе имеются команды РА и ПЗ:

– время между требованием прерывания и предоставлением прерывания (максимум) – 32 мкс,

– время между предоставлением прерывания и выборкой первой обслуживающей команды (максимум) – 18,8 мкс

– время между требованием прерывания и выборкой первой обслуживающей команды (максимум) – 50,8 мкс

4.3.10.4. При построении систем, в которых используются прерывания от внешних устройств, необходимо учитывать, что при выполнении всех команд РА и ПЗ периодически проверяется требование прерывания от внешних устройств. Если последнее установлено, процессор прекращает выполнение команды РА или ПЗ и возвращается в состояние, предшествующее выполнению этой команды. После выполнения программы обслуживания прерывания процесоор начинает выполнение прерванной команды РА или ПЗ сначала. Если внешние прерывания будут поступать слишком часто, то команда РА или ПЗ никогда не выполняется. Не следует вырабатывать сигнал требования прерывания с частотой свыше 3 кГц, если в программе используются команды РА или ПЗ.

4.3.11. Особенности выполнения некоторых команд

Пользователю, применяющему постоянные запоминающие устройства для хранения программы или самостоятельно изготовившему интерфейс для связи с устройством, необходимо учитывать следующее:

а) при выполнении команд CLR и CLRB к операнду приемника происходит обращение в цикле ВВОД–ПАУЗА–ВЫВОД;

б) при выполнении команды MOVB, обращение к операнду приемника выполняется в цикле ВВОД–ПАУЗА–ВЫВОД, в отличие от команды MOV , где обращение к операнду приемника выполняется в цикле ВЫВОД;

в) при выполнении команд MTRS и РА обращение к операнду

источника происходит в цикле ВВОД-ПАУЗА-ВЫВОД;

г) при выполнении команд ПЗ указатель стека должен указывать на оперативную память;

д) использование очистки разрешения прерывания (6 разряда) регистра какого-либо устройства может вызвать ошибку обращения к каналу и переход в режим связи с пультовым терминалом, если при выполнении команды, очищающей 6-й разряд, возникло требование прерывания от этого внешнего устройства. Для избежания подобной ситуации можно руководствоваться следующим примером:

```
РСП =0,     MTPS #200
CLR* #177564
MTPS #0.
```

4.4. Блок-схема центрального процессора

В этом подразделе рассматривается блок-схема ЦП, показывающая общую структуру ЦП на уровне функциональных блоков.

Основные функциональные блоки ЦП показаны на рис. 24 и описаны в последующих подразделах.

При изучении данного раздела необходимо руководствоваться *схемой электрической принципиальной* 3.858.382 Э3

4.4.I. Микропроцессорный набор БИС.

Основным функциональным блоком ЦП является микропроцессорный набор БИС (микропроцессор), состоящий из БИС управления, БИС регистрового арифметическо-логического устройства и двух БИС ПЗУ микрокоманд. Элементы набора связаны друг с другом 22-разрядным каналом микрокоманд МК 00-2IH.

Весь обмен информацией между микропроцессором и каналом ЭВМ осуществляется через БИС РАЛУ, а связь с линиями управляющих сигналов канала ЭВМ – через БИС управления.

Синхронизация работы всех БИС микропроцессора и других блоков ЦП обеспечивается четырьмя сериями неперекрывающихся тактирующих сигналов генератора (фазами Ф1, Ф2, Ф3 и Ф4).

Типичный цикл работы генератора составляет 400 нс (по 100 нс на каждый такт).

4.4.1,1. БИС управления

БИС управления осуществляет связь микропроцессора с линиями управляющих сигналов канала ЭВМ и генерирует последовательность адресов микрокоманд, по которым производится обращение к БИС МПЗУ. Выборка микрокоманда затем подается в БИС РАЛУ и БИС управления. Большинство микрокоманд выполняются в БИС РАЛУ, однако микрокоманды различных переходов, ветвлений выполняет БИС управления. Блок-схема БИС управления показана на рис. 25.

БИС управления содержит следующие функциональные блоки:

– программируемую логическую матрицу (ПЛМ), выполняющую дешифрацию команд и выработку адресов микрокоманд для МПЗУ,

– регистр микрокоманд, используемый для хранения текущей микрокоманды,

– счетчик микрокоманд, запоминающий адрес ячейки МПЗУ, из которой производится выборка,

– регистр возврата, используемый для запоминания адреса возврата при обращениях к микропрограммам.

– регистр команд, предназначенный для хранения текущей команды.

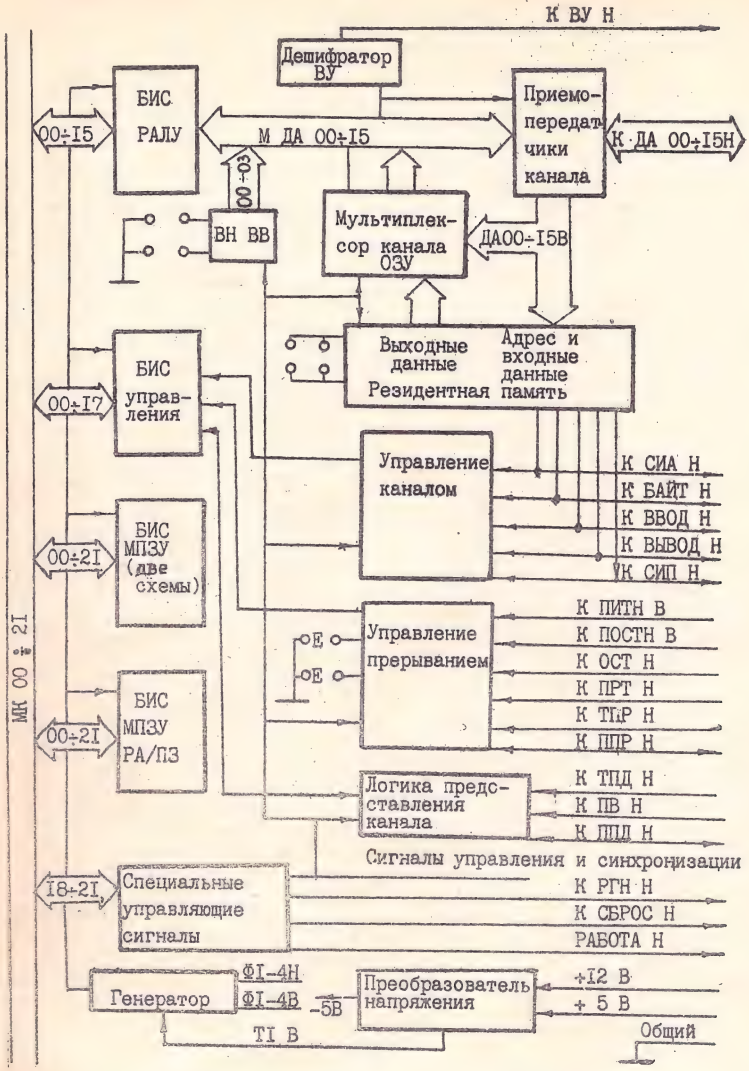


Рис. 24 Блок-схема ЦП

Рис. 25 Блок-схема БИС управления

3.858.382 ТО

– логические схемы управления каналом ЭВМ, которые вырабатывают управляющие и синхронизирующие сигналы.

– логические схемы управления прерыванием программы.

Связь микропроцессора с линиями управляющих сигналов канала ЭВМ осуществляется через блок управления каналом ЭВМ.

Генерация адресов МПЗУ определяется текущим содержимым счетчика микрокоманд, выполняемой командой, текущей микрокомандой, сигналами управления прерыванием и некоторыми другими специальными сигналами.

4.4.I.2. БИС регистрового арифметическо-логического устройства

БИС РАЛУ предназначена для выполнения арифметическо- логических преобразований над данными и осуществления информационного обмена с каналом ЭВМ. БИС РАЛУ снабжена для этого необходимыми средствами передачи, хранения и преобразования информации.

БИС РАЛУ содержит следующие функциональные блоки (рис. 26):

– блок обмена с микроканалом, через который осуществляется связь с другими элементами микропроцессорного набора.

– дешифратор адреса регистров общего назначения (РОН), обеспечивающий адресацию к 26 8-разрядным РОН.

– файл регистров общего назначения для хранения часто используемых данных.

– арифметическо-логическое устройство (АЛУ) осуществляющее арифметические и логические операции, необходимые для выполнения команд.

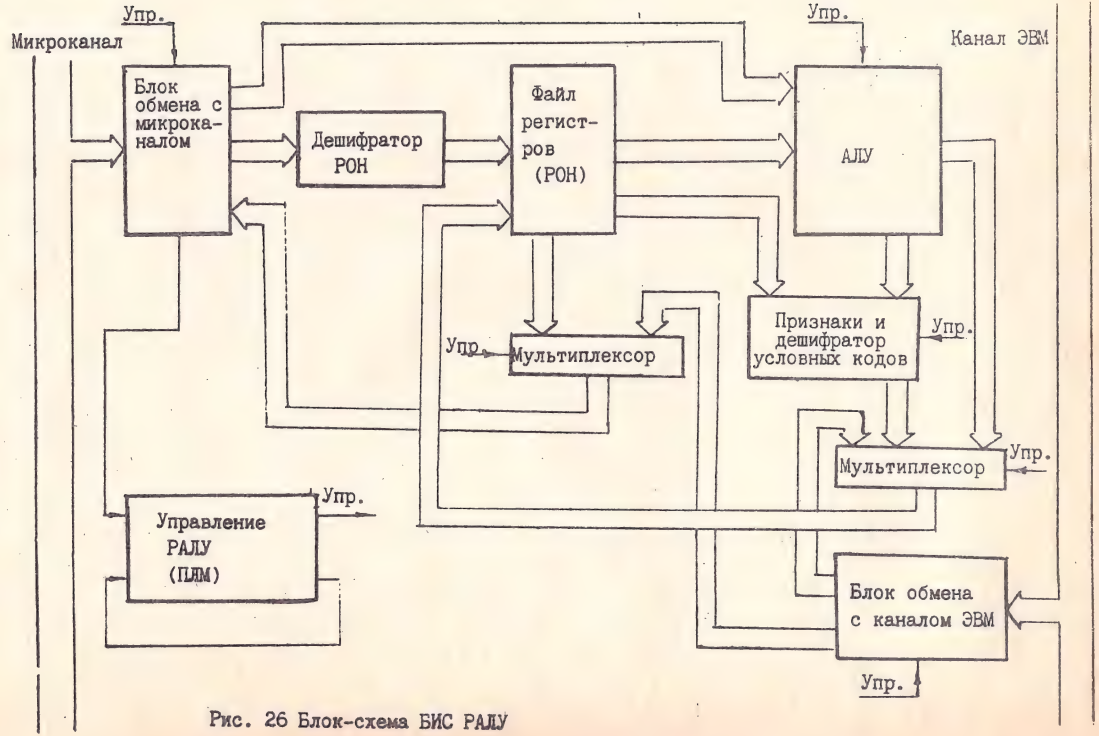


Рис. 26 Блок-схема БИС РАЛУ

– дешифратор команд условных переходов и логические схемы формирования признаков результата операции (кодов условий для команд ветвлений),

– блок обмена с каналом ЭВМ, осуществляющий связь микропроцессора с информационными линиями канала ЭВМ.

Файл регистров состоит из 26 8-разрядных регистров, IO из которых могут прямо адресоваться микрокомандами, 4 могут адресоваться либо прямо, либо косвенно, а оставшиеся I2 регистров могут адресоваться только косвенно. Косвенная адресация осуществляется с помощью специального трехфазного "У"регистра, входящего в состав блока обмена с микроканалом, который можно загружать из адресного поля микрокоманды.

I2 косвенно-адресуемых 8-разрядных регистров используется для реализации 6 регистров общего назначения (R0 ÷ R5). Четыре регистра, которые могут быть адресованы как прямо, так и косвенно (поскольку они осуществляют специальные функции процессора и к ним очень часто происходит обращение), содержат счетчик команд (СК) и указатель стека (УС).

Пять оставшихся пар прямо адресуемых регистров используются как рабочие регистры микропрограмм и обычно содержат: команду, исполнительный адрес (адрес операнда), первый операнд, второй операнд, слово состояния процессора и другую информацию о состоянии процессора.

8-разрядное АЛУ обрабатывает два операнда, адресованных микрокомандой. Когда задана операция над полным I6-разрядным словом, исполнение происходит двойными циклами, при этом во втором цикле инвертируется младший разряд адреса каждого регистра, содержащего операнд. Таким образом, I6-разрядный регистр

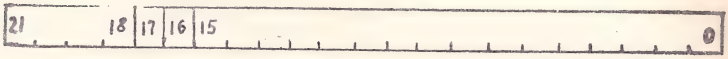
реализуется двумя последовательными 8-разрядными регистрами.

Пользователь имеет возможность программного доступа ко всем регистрам общего назначения (включая СК и УС) и к регистру состояния процессора.

4.4.I.3. БИС МПЗУ

БИС МПЗУ (см. рис. 27) хранит микрокоманды, эмулирующие систему команд ЭВМ "Электроника 60М", операции связи с пультовым терминалом и программу начального загрузчика.

Микрокомандное слово состоит из 22 разрядов.



I6 разрядов (с 0 по I5) содержат код микрооперации, разряд I6 используется для управления запоминанием адреса возврата из микроподпрограммы, разряд I7 инициирует выборку команды, остальные 4 разряда, с I8 по 2I, выполняют специальные управляющие функции. Следует заметить, что выходные сигналы разрядов I8 ÷ 2I являются ТТЛ - совместимыми.

4.4.2. Генератор тактирующих импульсов вырабатывает четыре серии неперекрывающихся импульсов с амплитудами I2 В и 3 В для осуществления синхронизации работы микропроцессора и различных блоков ЦП (см. рис. 26.)

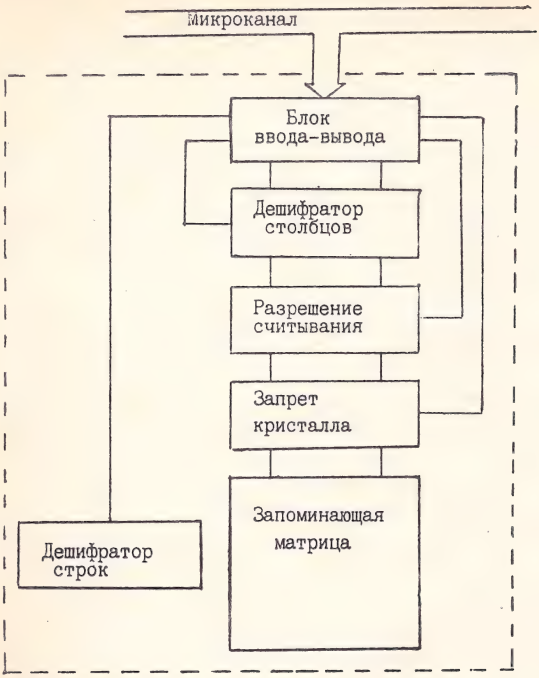


Рис. 27 Блок-схема БИС МПЗУ

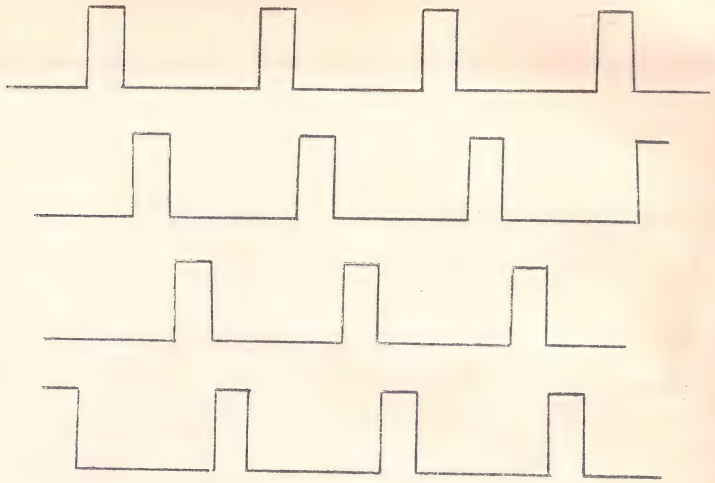


Рис. 28. Диаграмма тактирующих импульсов

ТТЛ – совместимые тактирующие импульсы Ф1 В, Ф2 В, Ф3 В и Ф4 В совпадают по форме и длительности с МФ1-4 В, но отличаются от них по амплитуде.

Управляющий осциллятор вырабатывает опорный сигнал с частотой, регулируемой в пределах от 10 до 13 МГц. Этот сигнал через вентили поступает на двухразрядный сдвигающий регистр. Выходные сигналы двухразрядного регистра расшифровываются дешифратором "1 из 4", и в результате получаются четыре основные фазы. Отрицательный импульс, вырабатываемый в момент переключения

дешифратора, запрещает его работу на ~ 10 нс, предотвращая перекрытие фаз. Каждый из четырех фазовых сигналов (М ФI÷М Ф4) является положительным, МДЛ – совместимым 90 нс (приблизительно) импульсом, передаваемым к каждому из БИС микропроцессорного набора.

Сигналы ФI Н÷Ф4 Н и ФI В÷Ф4 пофазно совпадают во времени они ТТЛ – совместимы и предназначены для синхронизации работы функциональных блоков ЦП.

4.4.3. Интерфейс канала и передача адреса/данных

Связь ЦП ЭВМ "Электроника 60М"с внешними устройствами и памятью осуществляется через канал с использованием 16 линий адреса/данных и 17 линий управления.

Для связи с канальными линиями К ДА в ЦП используются ИС приемо-передатчиков с открытым коллектором. Каждая ИС приемо-передатчика содержит четыре пары приемников и передатчиков. Каждый вход приемника соединен с выходом передатчика. Все четыре передатчика управляются одновременно двумя разрешающими сигналами. Схема одного разряда канального приемо-передатчика показана на рис. 29.

Управляющими сигналами для канальных приемо-передатчиков являются сигналы РЗП Н, СБРОС I В, ЦПД В.

Активный сигнал РЗП Н разрешает передачу в канал адреса или данных.

На рис. 30 приводится схема получения сигнала РЗП Н.

Приемо-передатчики управляющих сигналов канала запрещаются при начальной установке канала и в цикле прямого доступа к памяти.

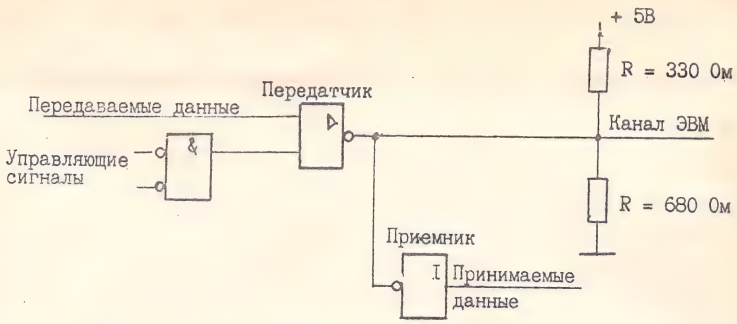


Рис. 29 Пример канального приемо-передатчика.

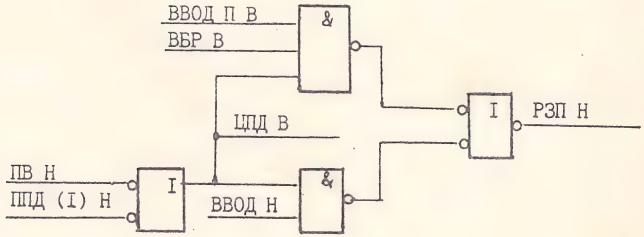


Рис. 30 Схема получения сигнала РЗП Н.

Каждая линия канала в ЦП соединяется с парой резисторов, которые обеспечивают 250 Ом импеданс канала и уровень "лог.0", равный 3,4 В.

4.4.4. Сигналы управления каналом

Сигналами управления каналом являются К СИА Н, К БАЙТ Н, К ВВОД Н, К ВЫВОД Н, К СИП Н. Логические схемы, вырабатывающие и передающие эти сигналы, показаны на рис. 31.

Сигнал синхронизации активного (управляющего каналом) устройства К СИА Н сообщает о том, что на линиях К ДА 00÷15 установлен адрес.

Задний фронт активного сигнала К СИА Н сигнализирует об окончании текущего цикла обращения к каналу. Процесс выработки сигнала К СИА Н начинается возбуждением сигнала М СИА В по синхроимпульсу Ф2 В в БИС управления. По заднему фронту Ф3 Н будет установлен триггер СИА. Установленный триггер СИА определяет выработку активного канального сигнала К СИА Н, который действует до окончания передачи информации, так как установленный сигнал СИП (I) В будет поддерживать триггер СИА в единичном состоянии. После того, как сигнал СИП (I) В станет пассивным, триггер СИА сбросится по заднему фронту Ф3 Н (сигнал М СИА В в это время будет уже пассивным).

Сигнал К БАЙТ Н используется для управления каналом в двух случаях.

Если К БАЙТ Н активен в момент установки сигнала К СИА Н, то он информирует канальные устройства о том, что начинается цикл ВЫВОД. Если он активен в момент установки сигнала К ВЫВОД Н, то он информирует об обращении к одному из байтов (в зависимости от состояния разряда К ДА 00 Н). Сигнал К БАЙТ Н

является инвертированным сигналом М БАЙТ В, который поступает с выхода БИС управления.

Сигнал К ВВОД Н используется в двух случаях работы канала. Во время активного сигнала К СИА Н он синхронизирует ввод данных в ЦП и требует отклика от управляемого (пассивного) устройства. Сигнал К ВВОД Н может также быть выработан для приема вектора прерывания. В этом случае К СИА Н вырабатываться не будет. Сигнал К ВВОД Н является инвертированным сигналом М ВВОД В, который вырабатывается БИС управления, когда ЦП готов к вводу информации.

Сигнал К ВЫВОД Н устанавливается, когда данные находятся на линиях К ДА и происходит вывод информации. БИС управления вырабатывает сигнал М ВЫВОД В по переднему фронту Ф2 В. Этот сигнал объединяется по "И" с пассивным СИП (I) Н, вырабатывая активный уровень на D - входе триггера ВЫВОД. Триггер ВЫВОД устанавливается по переднему фронту Ф3 В, инициируя активный сигнал К ВЫВОД Н.

Сбрасывается триггер ВЫВОД по Ф3 В сразу после того, как СИП (I) Н станет активным.

Сигнал синхронизации пассивного (управляемого) устройства К СИП Н устанавливается в ответ на К ВВОД Н или К ВЫВОД Н. Сигналы ВВОД Н и ВЫВОД Д (I) Н, объединяясь по схеме "ИЛИ", вырабатывают активный уровень сигнала В/В В всегда, когда происходит программная передача данных. Этот активный уровень разрешает работу счетчика ошибки обращения к каналу и выработку сигнала СИП В, который поступает на БИС управления. Триггер СИП устанавливается по переднему фронту Ф1 В и активный сигнал

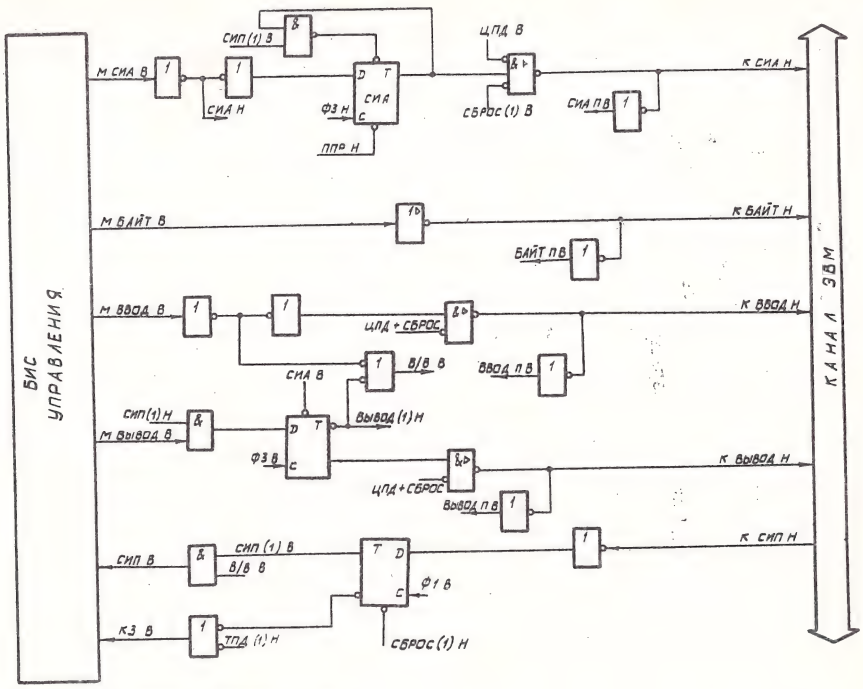


Рис. 31 Логика сигналов управления каналом.

СИП (I) Н, объединяясь по "ИЛИ" с ТПД(I) Н, вырабатывает активный сигнал КЗ В, который запрещает микропроцессору обращение к каналу ЭВМ до окончания действия сигнала СИП (I) Н. Поступающий на БИС управления сигнал СИП В информирует о том, что выводимые данные приняты, или что вводимые данные находятся в канале. Сигнал СИП В становится пассивным, когда становятся пассивными сигналы ВВОД Н или ВЫВОД Д(I) Н.

После этого пассивное устройство снимает сигнал К СИП Н, указывая на завершение своей части цикла передачи данных. По следующему синхроимпульсу ФI В триггер СИП очищается.

4.4.5. Дешифратор внешнего устройства.

Схема дешифратора показана на рис. 32. Дешифрация происходит в течение адресной части цикла канала. Вырабатываемый при этом сигнал К ВУ Н сигнализирует о том, что обращение идет к регистрам внешних устройств, адреса которых находятся в верхних 4К адресного пространства. Сигнал К ВУ Н запрещается во время передачи адреса вектора прерывания и во время цикла регенерации ОЗУ.

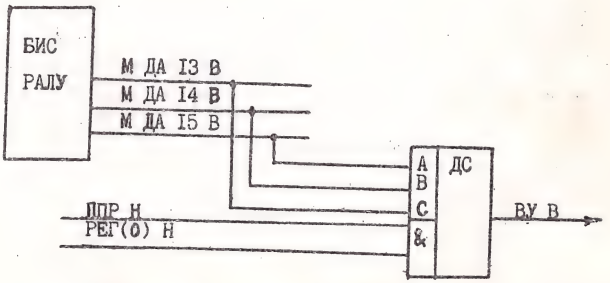


Рис. 32 Дешифратор внешнего устройства

3.858.382 ТО

4.4.6. Управление прерыванием и схема выработки сигнала СБРОС МК.

Логические схемы управления прерыванием и сброса счетчика микрокоманд показаны на рис. 32. Выработка сигнала СБРОС МК (I) Н обуславливается наличием ошибки по обращению к каналу или нарушением питания.

Выработка сигналов прерывания может инициироваться нарушением питания, регенерацией памяти, требованием прерывания по таймеру и требованием прерывания от внешнего устройства.

4.4.6.I. Нарушение и восстановление питания

Последовательность операций при нарушении питания начинается, когда сигнал К ПИТН В становится пассивным (низким), устанавливается триггер ПИТ МН. Сигнал с нулевого плеча этого триггера, объединяясь по "ИЛИ" с инвертированным сигналом ОСТ В, вырабатывает по Ф2 В активный уровень сигнала ТП ПИТ В.

Принимая этот сигнал, БИС управления прерывает выполнение программы и начинает выполнять микропрограммы обслуживания прерывания. Микропроцессор инициирует на линиях МК I8÷2I Н код I5, устанавливая тем самым триггер ВН ВВ. В цикле внутреннего ввода (ВН ВВ) микропроцессор принимает информацию с линий М ДА 00÷03 В о режиме пуска, наличии сигнала ошибки по обращению к каналу и сигнала ПИТ МН В (питание меньше нормы). Активный сигнал ПИТ МН В информирует его о том, что прерывание программы вызвано нарушением питания, а не требованием останова. Если процесс нарушения питания продолжается, К ПОСТ Н В становится пассивным (низкий уровень), вырабатывая активный сигнал ПОСТ МН Н. Этот сигнал очищает триггеры ПИТ МН, требований прерывания, вырабатывает сигнал СБРОС МК (I) Н и устанавливает в исходное состояние все подключенные к каналу устройства.

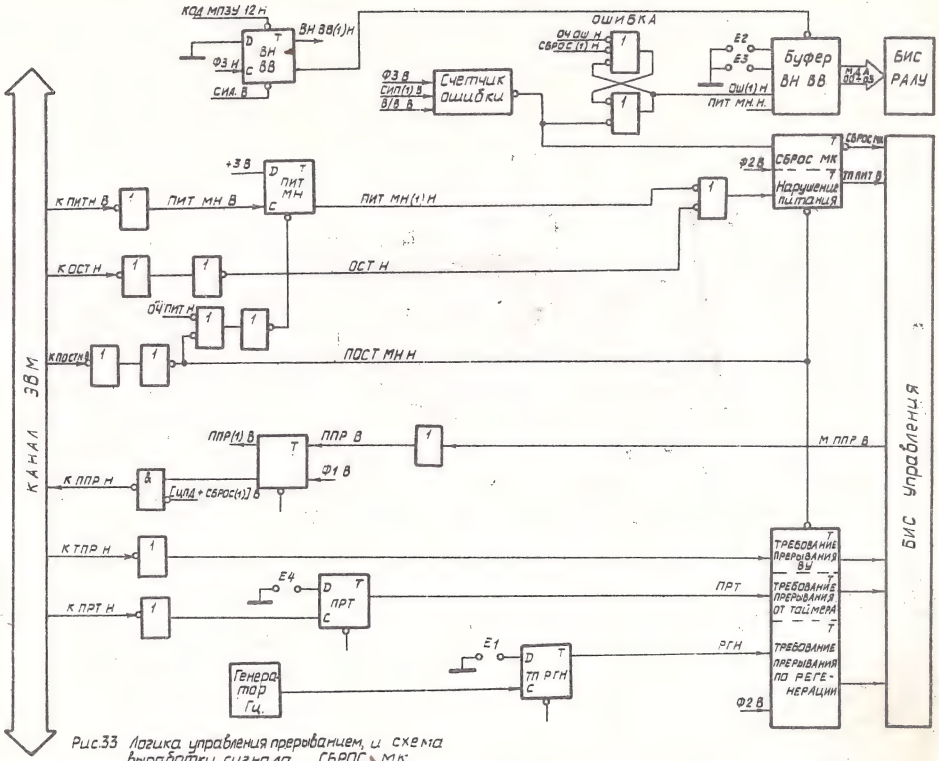


Рис.33 Логика управления прерыванием, и схема выработки сигнала СБРОС МК.

Активный сигнал СБРОС МК (I) Н сбрасывает счетчик микрокоманд в БИС управления, переводя микропроцессор в состояние "нет операции". Микропроцессор выходит из этого состояния, когда сигнал К ПОСТ Н снова станет активным.

При восстановлении питания сигнал ПОСТ МН Н становится пассивным и по следующему синхроимпульсу Ф2 В снимает сигнал СБРОС МК (I) Н. Микропроцессор отвечает выполнением цикла ВН ВВ для определения, какая микропрограмма пуска выбрана с помощью перемычек. По окончании цикла микропроцессор выполняет эту микропрограмму и, когда вырабатывается К ПИТН В, возобновляется его нормальная работа. Следует заметить, что микропрограмма восстановления питания включает в себя первоначальную установку канальных устройств (выработка сигнала К СБРОС Н).

4.4.6.2. Режим связи с пультовым терминалом (останов).

Вход в режим останова осуществляется либо при выполнении команды HALT, либо при выработке каким-либо канальным устройством сигнала К ОСТ Н (например, панелью управления). Процессор прекращает выполнение программы и переходит на выполнение микропрограммы, как было описано для случая нарушения питания. Однако, когда процессор выполняет цикл ВН ВВ, сигнал ПИТ МН В (М ДА ОЗ В) пассивен, и он начинает выполнять микропрограмму связи с пультовым терминалом (а не микропрограмму обслуживания падения питания).

Снятие сигнала К ОСТ Н позволит процессору возобновить работу по выполнению программы (при условии выдачи соответствующей команды с пультового терминала).

По следующему импульсу Ф2 В сигнал ТП ПИТ В становится низким и разрешается режим программной работы.

4.4.6.3. Ошибка обращения к каналу

Ошибки, возникающие при обращении к каналу, вызывают прекращение выполнения программы и переход на подпрограмму обслуживания внутреннего прерывания программы через вектор с адресом 4. Ошибка возникает, если устройство в результате неисправности не отвечает на сигналы процессора К ВВОД Н или К ВЫВОД Н сигналом К СИП Н в течение приблизительно 10 мкс.

Это обеспечивается разрешением работы счетчика времени ожидания ответа, который запускается при каждом обращении к каналу.

Если циклом канала управляет другое канальное устройство (не ЦП), работа счетчика будет запрещена. Счетчик работает от импульсов ФЗВ. При нормальной работе вырабатывается активный сигнал К СИП Н, запрещающий работу счетчика.

Однако, если К СИП Н не поступает в центральный процессор в течение 10 мкс, счетчик достигает своего максимального значения, что и является критерием появления ошибки. Сигналы ОШ Н и ОШ (1) В становятся активными, и по следующему импульсу Ф2 В вырабатывается активный сигнал СБРОС МК (1) Н, который поступает в БИС управления. Микропроцессор отвечает выполнением микропрограммы "начальной установки". В начале ее он выполняет цикл ВН ВВ и определяет, что возникла ошибка по обращению к каналу, а не нарушение питания. Затем процессор переходит на выполнение программы обслуживания прерывания по ошибке обращения к каналу. Для очистки триггера ОШ вырабатывается сигнал ОЧ ОШ Н, соответствующий коду 12 на линиях МК 18÷21 Н.

4.4.6.4. Прерывания от внешних устройств. Требования прерывания от внешних устройств поступают по линии К ТПР Н. Этот сигнал инвертируется и вырабатывает по импульсу Ф2 В активный

уровень ТП ВУ, который информирует процессор о требовании прерывания. Если приоритет процессора в слове состояния равен 0, он отвечает выработкой сигналов М ПРВ и М ВВОД В. Через буфер сигнал М ВВОД В поступает на линии К ВВОД Н, информируя устройства о том, что они должны подготовить к работе логику определения приоритета по отношению к другим требующим прерывание устройствам (логику передачи сигнала К ПР Н).

В следующем цикле после выработки сигнала К ВВОД Н инвертированный сигнал М ПР В установит по заднему фронту импульса ФI Н триггер ПР (предоставление прерывания). В результате этого сигнал К ПР Н станет активным.

Устройство, имеющее самый высокий приоритет среди требующих прерывание, отвечает на процессорные сигналы К ВВОД Н и К ПР Н тем, что помещает свой адрес вектора прерывания на линии К ДА и передает К СИП Н, вызывая ввод адреса вектора в процессор.

Причем, К СИА Н не вырабатывается во время этой операции и не производится адресация какого-либо устройства.

Устройство, требовавшее прерывание, снимает свой сигнал К ТПР Н. Процессор отвечает на К СИП Н снятием своих сигналов К ВВОД Н и К ПР Н и начинает выполнение подпрограммы обслуживания прерывания со стартовым адресом, который хранится в указанном векторе.

4.4.6.5. Прерывание по регенерации динамической памяти

Регенерация является необходимым условием сохранения информации в динамической памяти на МДП БИС. Процесс регенерации начинается по сигналам 500 Гц генератора, причем перемычка ЕП должна быть удалена.

По переднему фронту синхроимпульса регенерации СИ РГН В устанавливается триггер требования регенерации. По следующему импульсу Ф2 В вырабатывается сигнал ТП РГН В, который поступает на БИС управления.

Микропроцессор отвечает выработкой активного сигнала УСТ РГН Н, соответсвующего коду I3 на линиях МК I8÷2I, и начинает выполнение микропрограммы регенерации памяти. Триггер РГН, инициирующий активный уровень сигнала К РГН Н устанавливается, а триггер требования регенерации сбрасывается.

По окончании микропрограммы регенерации триггер РГН очищается кодом I2 с линий МК I8÷2I.

Выработка сигнала К РГН Н будет запрещена, если в данный момент имеет место цикл прямого доступа к памяти или происходит операция по первоначальной установке канальных устройств.

4.4.6.6. Прерывание по таймеру

Прерывание по таймеру обычно используется устройствами, требующими обслуживания прерывания через заданные промежутки времени. Инициатором прерывания в этом случае является таймер (часы реального времени).

Однако, если в системе отсутствуют устройства, которые должны работать в реальном масштабе времени, пользователь может использовать линию К ПРТ Н для своих целей.

Прерывание по таймеру отличается от прерывания от внешнего устройства более высоким приоритетом и тем, что не требуется адрес вектора для того, чтобы начать подпрограмму обслуживания прерывания.

Обслуживание начинается через вектор с адресом 100_8, который зарезервирован специально для прерывания от таймера.

Процесс предоставления прерывания начинается установкой

сигнала К ПРТ Н. Этот сигнал устанавливает в ЦП триггер ПРТ.

Установка этого триггера будет разрешена, если отсутствует перемычкаЕ13. По следующему импульсу Ф2 В вырабатывается активный сигнал ТП ТМ В, который поступает на БИС управления. Если приоритет процессора в слове состояния равен 0, прерывание может быть выполнено.

Требование прерывания по таймеру сбрасывается выработкой процессорного сигнала ОЧ ТМ Н.

4.4.7. Специальные управляющие сигналы.

Специальные управляющие сигналы предназначены для выполнения операций по первоначальной установке канальных устройств, регенерации памяти, а так же для реализации пяти внутренних операций центрального процессора.

Схема выработки специальных управляющих сигналов показана на рис. 34.

Буфер, выполненный на стандартных ТТЛ вентилях, преобразует сигналы с линий микроканала МК I8÷2I в сигналы со стандартными ТТЛ уровнями Б КОД 0÷3 В.Этот четырехразрядный код поступает на дешифратор, работа которого будет разрешена лишь в том случае, если сигнал Б КОД 3 В активный. Кроме того, работа дешифратора стробируется синхроимпульсом Ф3 В.

Выходы дешифратора поступают коды I0÷I7.

Ниже приведено краткое описание функций специальных управляющих сигналов.

Код I0 - не используется.

Код II (ОЧ СБРОС Н или РАБОТА Н) - этот код вырабатывается микропроцессором для очистки триггера СБРОС и установки сигнала РАБОТА Н, который используется для индикации режима программной работы на передней панели ЭВМ "Электроника 60М".

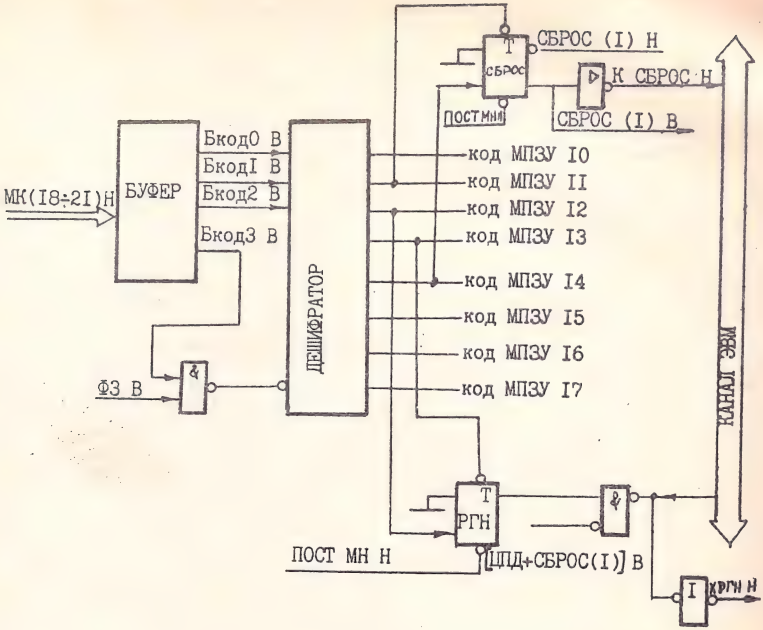


Рис. 34 Блок-схема выработки специальных управляющих сигналов

Код I2 (ОЧ ОШ Н) – этот код является сигналом сброса для триггеров регенерации (РГН) и ошибки по обращению к каналу (ОШ).

Данная функция завершает прерывание по регенерации и по ошибке обращения к каналу.

Код I3 (УСТ РГН Н) – это код используется для установки триггера РГН (регенерация). Активный сигнал с единичного плеча этого триггера инициирует выработку активного сигнала К РГН Н.

Триггер РГН обычно сбрасывается выработкой микропроцессором сигнала ОЧ ОШ Н (код I2) после завершения операции регенерации.

Однако он может быть очищен и при нарушении питания (ПОСТ МН Н становится активным и очищает триггер).

Код I4 (первоначальная установка канала) – Программная первоначальная установка канала может быть выполнена исполнением команды RESET . Микропроцессор отвечает выработкой кода I4 в результате чего триггер СБРОС установится в единичное состояние.

Установленный триггер будет инициировать выработку активного сигнала К СБРОС Н.

Примерно через I0 мкс микропроцессор вырабатывает сигнал ОЧ СБРОС Н, очищая триггер СБРОС.

Кроме того, триггер первоначальной установки СБРОС может быть установлен при нарушении питания сигналом ПОСТ МН Н. При восстановлении питания стартовая микропрограмма очистит триггер СБРОС сигналом ОЧ СБРОС Н.

Код I5 (цикл внутреннего ввода) – этот код вырабатывается процессором для установки триггера ВН ВВ, начиная цикл внутрен-

него ввода. Этот цикл позволяет микропроцессору принять дополнительную информацию о режиме пуска по восстановлению питания о наличии сигнала ошибки по обращению к каналу или сигнала нарушения питания. В зависимости от вида принятой информации микропроцессор выбирает тот или иной путь выполнения микропрограммы. Таким образом, принимаемая информация является своеобразным вектором, по которому происходит ветвление микропрограммы.

Код I6 (ОЧ ПИТ) – этот код вырабатывается процессором для очистки триггера ПИТ МН.

Код I7 (ОЧ ТМ) – этот код вырабатывается процессором для очистки триггера требования прерывания по таймеру.

4.4.8. Логика предоставления канала.

Логика предоставления канала обеспечивает использование канала как центральным процессором, так и устройствами прямого доступа к памяти (устройвами ПДП).

Пока устройства ПДП не требуют канал, обмен данными происходит программно. Но когда устройство ПДП становится управляющим устройством процессор приостанавливает свои операции до тех пор, пока операция ПДП не будет закончена.

При программном обмене управляющим устройством является центральный процессор и триггер требования прямого доступа (ТПД) очищен (см. рис. 35). Очищен также триггер разрешения прямого доступа (РПД).

Устройство, вырабатывающее требование прямого доступа к памяти, устанавливает активный сигнал на линию К ТПД Н.

Этот сигнал, поступая в центральный процессор, устанавли-

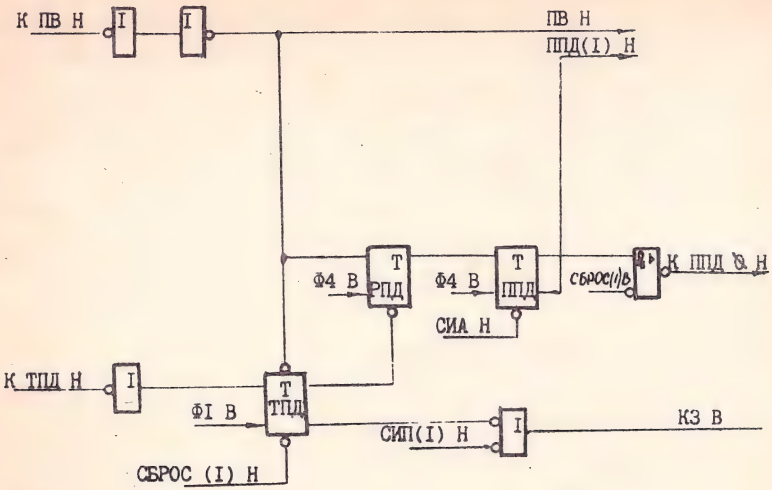


Рис. 35 Логика предоставления прямого доступа к памяти

вает по синхроимпульсу Ф1 В триггер ТЦД. Активный уровень с нулевого плеча этого триггера, объединяясь по "ИЛИ" с СИП (I) Н, инициирует сигнал КЗ В (канал занят) и вынуждает центральный процессор ожидать окончания цикла ПДП. Активный уровень с единичного триггера ТЦД устанавливает по синхроимпульсу Ф4 В триггер РПД. Затем логика предоставления канала, ожидает когда процессор, заканчивая текущий цикл обращения к каналу, очистит линию К СИА Н. По первому синхроимпульсу Ф4 В, следующему за пассивным К СИА Н, устанавливается триггер предоставления прямого доступа к памяти (ПДП), инициируя активный сигнал К ПДП Н.

Активный сигнал ПДП (I) Н запрещает работу приемо-передатчиков, связывающих линии КДА 00-I5 Н с микропроцессором, и восьми передатчиков сигналов управления каналом. Устройство ПДП в ответ на сигнал К ПДП Н устанавливает активный уровень сигнала на линии К ПВ Н и очищает линию К ТЦД Н. Принятый центральным процессором сигнал подтверждения выбора ПВП Н поддерживает триггер ТЦД в единичном состоянии, а триггер РПД очищает. По следующему синхроимпульсу Ф4 В после очистки триггера РПД очищается триггер ПДП и сигнал К ПДП Н становится пассивным. Сигнал К ПВ Н останется активным на протяжении всей операции ПДП, предотвращая удовлетворение нового требования прямого доступа к памяти.

Устройство ПДП освобождает канал, сбрасывая сигнал К ПВ Н.

Следующий синхроимпульс Ф1 В сбрасывает триггер ТЦД. Затем сигнал КЗ становится пассивным, разрешая процессору

начать свой цикл обращения к каналу. Начинающийся процессорный цикл обращения к каналу, вырабатывая К СИА Н, запрещает установку триггера ШД на все время процессорного цикла.

4.4.9. Блок резидентной памяти.

Центральный процессор имеет оперативное запоминающее устройство емкостью 4К I6-разрядных слов. Резидентная память может занимать в адресном пространстве системы положение нулевого или первого банка. Для выбора требуемого банка (0 или I) на модуле должна быть установлена одна из двух специальных перемычек. Основные функциональные блоки резидентной памяти показаны на рис. 36. Она состоит из I6 корпусов МДП БИС с организацией 4К х I бит и логики адресации и управления. Ниже дается подробное описание выполнения операций адресации, записи и чтения. Адресация начинается управляющим устройством (центральным процессором или устройством ПДП), помещающим I6-разрядный адрес на линии К ДА 00÷I5 Н. Следует отметить, что адрес, относящийся к ячейке резидентной памяти, появляется на линиях К ДА 00÷I5 Н даже, если управляющим устройством является центральный процессор.

Адрес, переданный в канал, защелкивается в регистре адреса. Для выбора банка памяти используются I3-I5 разряды, для адресации ячейки памяти используются разряды с 0I по I2,0 разряд указывает на байт, к которому идет адресация при байтовых операциях.

При совпадении адреса банка памяти с адресом, выбранным перемычками, или при регенерации памяти вырабатывается сигнал ВБР В, который является разрешающим для логики управления.

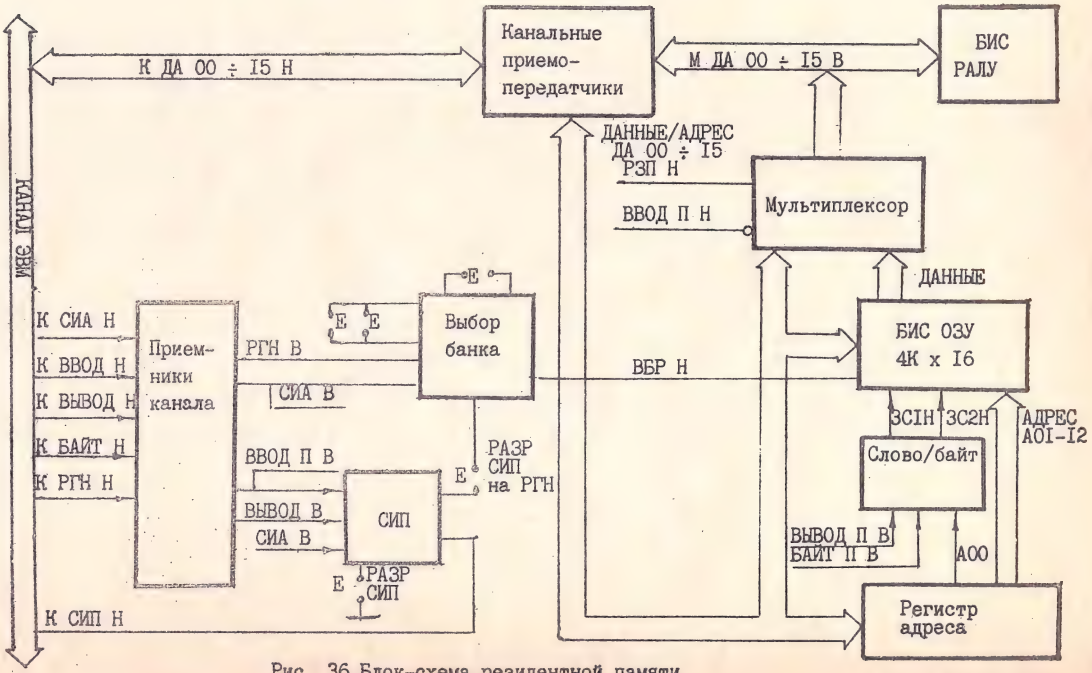


Рис. 36 Блок-схема резидентной памяти

В течение цикла ВВОД резидентная память принимает адрес с системного канала, примерно через 100 нс после появления сигнала К ВВОД Н ОЗУ передает сигнал К СИП Н и подготавливает данные, которые мультиплексируются на БИС РАЛУ минуя канал, если не выполняется цикл ПДП.

В цикле ВЫВОД после адресной части, которая аналогична адресной части цикла ВВОД, в канал помещаются данные и устанавливается сигнал К БАЙТ Н для байтовых операций. Если этот сигнал пассивен, то происходит запись всего слова, если активен, то при четном адресе происходит запись младшего байта, а при нечетном – старшего.

При операции СЧИТЫВАНИЕ-МОДИФИКАЦИЯ-ЗАПИСЬ осуществляется считывание данных из адресуемой ячейки памяти, модификация этих данных (т.е. выполнение операции в АЛУ) и запись в ту же ячейку памяти.

Эта операция выполняется за один цикл ВВОД-ПАУЗА-ВЫВОД обращения к каналу. Приблизительно через каждые 2 мс памяти на динамических МДПБИС требуется проводить операцию регенерации. Она проводится под управлением либо микропрограммы процессора, либо устройства ПДП.

Операция регенерации заключается в выполнении 64 циклов ВВОД при адресации по строкам.

Операция регенерации начинается управляющим устройством установкой активного сигнала К РГН Н.

Затем управляющее устройство последовательно выполняет операцию К СИА Н/К ВВОД Н при наращивании содержимого разрядов адреса А 1-6 после каждого цикла.

Операция регенерации производится во всем объеме памяти одновременно. Резидентная память на каждый активный К ВВОД Н при регенерации должна отвечать установкой К СИП К.

4.4.IO. Преобразователь постоянного напряжения питания.

Эта схема обеспечивает выработку непосредственно в ЦП отрицательного напряжения –5 В. Постоянное напряжение питания на схему преобразователя подается от источника +I2 В. Частота переключения преобразователя определяется выходным сигналом Т I В генератора импульсов с частотой 2,8 МГц. Стабализированное напряжение –5 В подается на все БИС ОЗУ и микропроцессорного набора.

4.5. Режим работы ЦП

На модуле ЦП имеются перемычки, при помощи которых пользователь может выбрать тот или иной режим работы. В табл. 17 приводится список перемычек и функций, реализуемых с их помощью.

Эти функции следующие:

запрет прерывания по таймеру; выбор режима пуска при включении питания; запрет регенерации памяти; выбор номера банка резидентной памяти; запрещение СИП в течение цикла регенерации от резидентной памяти; запрещение СИП от резидентной памяти; выбор напряжения СМ1 и СМ2. На предприятии-изготовителе перемычки устанавливаются в соответствии с табл. 18. Для конкретного применения пользователь может изменить их установку, как это описано в последующих подразделах.

Таблица I7

Перемычка	Положение перемычки	Функции
E1 + E4		Выбор напряжения источника отрицательного смещения для БИС
E5		Управление формирователем тактовых импульсов
E6		Выбор режима пуска
E7		Управление формирователем тактовых импульсов
E8	0/I	Резидентная память запрещена/разрешена
E9,E10		Выбор номера куба резидентной памяти
E11	0/I	Регенерация памяти разрешена/запрещена
E12		Выбор режима пуска
E13	0/I	Прерывание по таймеру разрешено/запрещено
E14	0/I	СИП от резидентной памяти при регенерации разрешен/запрещен
E15	0/I	СИП от резидентной памяти разрешен/запрещен
E16	I/0	
E17,E18		Выбор напряжения источника отрицательного смещения для БИС

Примечание: I – перемычка установлена, 0 – перемычка удалена.

Режимы пуска

E6	E12	
0	0	Пуск через 24 вектор
0	I	Выход на микропрограмму связи с пультом

I	0	Выход на ячейку I73000
I	I	Переход на резервную микрокоманду

Номер куба резидентной памяти

E9	EI0	
0	I	Выбран куб 0
I	0	Выбран куб I

Таблица 18

№ № п/п	Положение перемычки	Перемычки
I.	Установлена	E5 E7, EI0, EI5, E6
2.	Удалена	EI2, E9, EII, EI3 EI4, EI6
3.	Устанавливается при настройке в соответствии с паспортом БИС	EI – E4, EI7, EI8

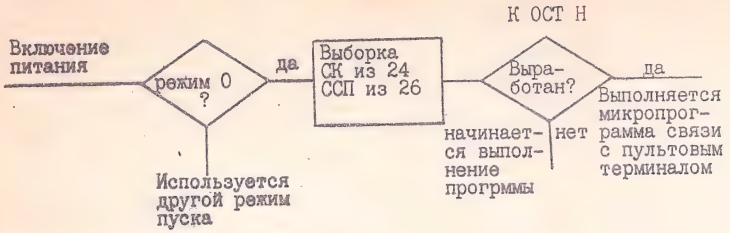


Рис. 37 Последовательность операции при пуске после включения питания в режиме 0.

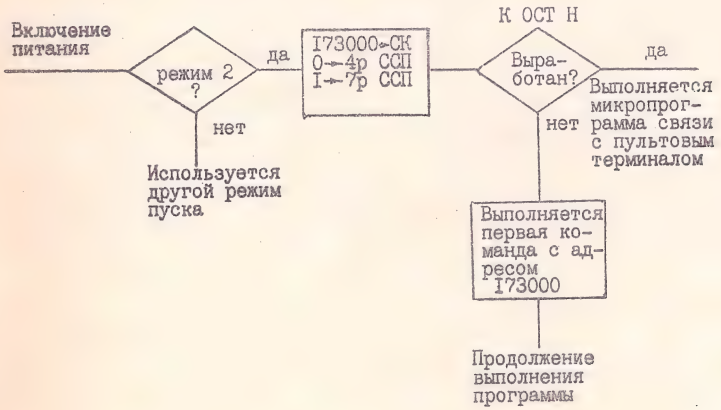


Рис. 38 Последовательность операций при пуске после включения питания в режиме 2.

4.5.1. Регенерация памяти.

ЦП способен управлять регенерацией всей динамической МДП-памяти в системе, если перемычка ЕII удалена.

Регенерация памяти нужна в том случае, если ЦП работает с динамической памятью, будь то резидентная память или модули ОЗУ емкостью 4К I6-разрядных слов. Операция регенерации может проходить под управлением и другого устройства, например, устройства управления регенерацией в режиме ПДП. Если такое устройство используется или в системе не используется динамическая память, перемычка ЕII должна быть установлена.

Если последовательность операций по регенерации выполняется под управлением центрального процессора, то через каждые 2 мс вырабатывается требование прерывания по регенерации.

Процесс регенерации занимает около I50 мкс, в течение которых внешние прерывания (К ТПР Н и К ПРТ Н) игнорируются.

Однако, запросы ПДП могут удовлетворяться в промежутках между выполнением циклов К СИА Н/К ВВОД Н, вырабатываемых во время процесса регенерации.

4.5.2. Таймер с запуском от сети.

Прерывание программы по таймеру (или внешнему событию) разрешено, если перемычка ЕI3 удалена и процессор находится в режиме программной работы.

Чтобы запретить прерывание, перемычка должна быть установлена. Прерывание инициируется таймером или другим внешним устройством, которые вырабатывают К ПРТ Н. Первая команда программы обслуживания выбирается обычно в пределах 40 мкс с момента выработки сигнала К ПРТ Н. Это время может увеличиться за счет выполнения внутреннего прерывания (регенерация памяти, прерывание по Т-разряду, нарушение питания и т. д.)или при выработке сигнала К ОСТ Н.

4.5.3. Выбор режима пуска при включении питания.

Поскольку ЦП может использоваться в различных системах, в том числе с памятью без сохранения информации при отключении питания(полупроводниковое ОЗУ), либо с сохранением информации (ПЗУ, МОЗУ), пользователь имеет возможность выбрать один из четырех режимов пуска при включении питания.

Эти режимы задаются перемычками ЕI2 и Е6. Следует отметить, что перемычками определяются только режимы пуска при включении или восстановлении питания, когда сигналы К ПИТН В и К ПОСТ НВ

становятся активными. При выключении питания перемычки Е6 и Е I2 на работу ЦП влияния не оказывают.

Для выполнения последовательности операций при включении питания имеет значение состояние сигнала К ОСТ Н.

Если он активен, вызывается выполнение микропрограммы связи с пультовым терминалом.

Подробное описание каждого режима пуска дается далее.

Режим 0.

В этом режиме процессор выполняет последовательность микрокоманд, которые извлекают содержимое 24-й и 26-й ячеек памяти и загружают его соответственно в СК и РСП. Затем микропрограмма проверяет состояние сигнала К ОСТ Н.

В зависимости от его состояния процессор либо переходит к выполнению микропрограммы связи с пультовым терминалом (К ОСТ Н – низкий уровень), либо начинает выполнение программы, используя текущее содержимое СК в качестве стартового адреса.

Следует отметить, что Т-разряд (4 разряд РСП) загружается содержимым 4-го разряда из ячейки 26.

Описываемый режим следует использовать только в том случае, если память не теряет информацию при выключении питания или, если при включении питания вырабатывается активный сигнал К ОСТ Н. Последовательность операций для этого режима показана на рис. 37.

Режим I.

В этом режиме процессор сразу переводится на выполнение микропрограммы связи с пультовым терминалом, независимо от состояния сигнала К ОСТ Н.

При этом предполагается, что интерфейс пультового терминала имеет регистр состояния с адресом I77560.

Режим 2.

В этом режиме процессор выполняет микропрограмму, которая загружает в СК стартовый адрес I73000 и, если не вырабатывается сигнал К ОСТ Н, начинает выполнение программы. Заметим, что в СК загружается адрес I73000, в 4-й разряд РСП (Т-разряд) заносится 0, а разряд 7 (запрет прерывания) - устанавливается. Программа пользователя должна установить или очистить эти разряды, как это ей необходимо, и установить правильное значение указателя стека (R6).

Этот режим следует использовать, когда ячейка с адресом I73000 принадлежит памяти, не теряющей информации при включении питания (ПЗУ, ППЗУ или МОЗУ).

Если в системе нет устройства с таким адресом, то произойдет прерывание программы по вектору с адресом 4.

Если вырабатывается К ОСТ Н, процессор не будет выполнять команду из ячейки I73000, а сразу же начнет выполнять микропрограмму связи с пультовым терминалом.

Последовательность операций для этого режима пуска показана на рис. 38.

Режим 3.

Последовательность микрокоманд, выполняемая в этом режиме, позволяет обращаться к той области МПЗУ, которая зарезервирована под микропрограммы пользователя, расширяющие возможности процессора (четвертая страница МПЗУ, ячейки3000-3777).

После выработки сигналов К ПОСТН В и К ПИТН В и очистки

внутренних флагов. выполняется микроветвление на микроячейку 3002.

Если выбран этот режим и в четвертой странице нет МПЗУ, будет выполение микропрерывания через микроячейку 0000, которое, в свою очередь, вызовет прерывание по попытке выполнить команду с резервным кодом операции (по вектору с адресом 10).

Следует отметить, что до передачи управления четвертой странице МПЗУ состояние сигнала К ОСТ Н не проверяется.

4.5.4. Выбор адреса банка резидентной памяти.

Перемычки Е9 и Е10 используются для выбора адреса банка (4К) резидентной памяти. Причем устанавливается только одна перемычка. Если установлена перемычка Е9, адреса резидентой памяти воспринимаются как адреса первого банка (ячейки 20000 - 37776). Если же установлена перемычка Е10, резидентной памяти соответствуют адреса нулевого банка (000000-17776).

4.5.5. Запрещение СИП от резидентной памяти.

Перемычку Е14 следует удалить, если необходимо запретить СИП от резидентной памяти в течение цикла регенерации.

Перемычку Е15 следует удалить, а перемычку Е16 установить, если необходимо запретить СИП от резидентной памяти.

4.6. Команды пультового терминала.

ЭВМ не имеет специального пульта управления, при помощи которого оператор получает возможность выполнять различные операции управления (занесение адреса, считывание данных, запись данных и др.). Все эти функции выполняются устройствами, способными передавать в ЦП и принимать из ЦП коды ГОСТ 13052-74, соответствующие буквенно-цифровым символам. Принимаемые из ЦП коды должны быть интерпретированы устройством таким образом, чтобы оператор имел возможность контролировать работу процессора. Такое устройство принято называть пультовым терминалом. В режим связи с пультовым терминалом ЦП может войти при вклю-

чении питания или перейти в него из режима программной работы в следующих случаях:
- после выполнения команды HALT;
- при двойной ошибке обращения к каналу;
- при ошибке во время регенерации памяти;
- при ошибке во время передачи вектора прерывания;
- при установке активного сигнала на линии К ОСТ Н.

Если при включении питания ЦП выходит в режим пультового терминала, ЭПМ выводит на печать 6-разрядный восьмеричный код, который определяется заданием одного из четырех режимов пуска.

При переходе из режима программной работы в режим связи с пультовым терминалом ЦП через ЭПМ выполняет следующую последовательность операций: возврат каретки (ВК), перевод строки (ПС), вывод на печать адреса следующей команды, которая должна быть исполнена, повторное выполнение возврата каретки (ВК), перевод строки (ПС) и вывод на печать знака @ . Знак@ означает готовность ЦП принять команду оператора.

В случае, если оператор вводит неправильные команды, ЭПМ печатает знак ?, выполняет ВК,ПС и печатает знак@, то есть команду необходимо ввести повторно.

Печать выглядит следующим образом:

@неправильная команда? (ВК), (ПС)
@

Заметим, что выводимые на ЭПМ адреса и данные всегда имеют 6 восьмеричных разрядов. Если при вводе 6-разрядного восьмеричного кода в ЦП старшие разряды равны нулю, то для экономии времени требуемый код можно начинать вводить со старшего, отличного от нуля, разряда. Например,

вместо 000020_8

допускается 20_8.

Ниже приводятся и описываются команды, подаваемые с пультового терминала.

I. Команда "/" (код 057) - "открыть ячейку".

По этой команде распечатывается содержимое ячейки памяти, регистра общего назначения или слова состояния процессора. Она обычно следует за адресом ячейки.

Пример:

@ I000/02I525

где I000 - восьмеричный адрес ячейки памяти, которая должна быть раскрыта,
02I525 -содержимое раскрытой ячейки с адресом I000.

Если команда "/" используется без задания адреса, она открывает содержимое последней ячейки, которая открывалась ранее. Это используется для проверки только что записанной информации.

В приведенном выше и последующих примерах выводимые из ЦП буквенно-цифровые символы подчеркнуты.

2) Команда "ВК" (код 0I5) - "закрыть ячейку".
Если требуется изменить содержимое открытой ячейки, команда "ВК" подается после того, как оператор напечатал новое содержимое ячейки.

Если не требуется менять содержимое, команда "ВК" должна следовать сразу за содержимым ячейки, которое напечатала ЭВМ.

Пример

```
I)@00I000/0I2525 "ВК" (ВК, ПС)
  @
2)@00I000/0I2525     I2642I "ВК" (ВК, ПС)
  @00I000/I2642I
```

где команда "ВК" используется в обоих примерах для закрытия ячейки с адресом I000. Во втором примере в ячейке с адресом I000 информация изменена оператором. В скобках указаны действия, выполняемые ЭПМ под управлением ЦП.

3) Команда "ПС" (код 0I2) - "закрыть ячейку и открыть следующую". Она открывает ячейку, адрес которой определяется содержимым СК. Эта команда используется для распечатки содержимого массивов последовательно расположенных ячеек. Если содержимое открытой ячейки (или РОНа) должно быть изменено, то команда подается после того, как оператор напечатал новое содержимое ячейки.

Пример.

```
@00I000/0I2525  "ПС" (ПС, ВК)
@00I002/005252        (ВК, ПС)
                "ВК"
@
```

Команда "ПС" закрывает ячейку с адресом I000 и открывает ячейку с адресом I002.

Если команда "ПС" используется при открытом РСП, она за-

крывает ячейку, не открывая следующей.

Если команда "ПС" используется применительно к РОН с номером 7, она закрывает его и распечатывает содержимое R0.

4) Команда "←" (код 136) - "открыть предыдущую ячейку". Она закрывает ранее открытую ячейку и открывает ячейку с адресом, уменьшенным на 2 или (для РОН) - уменьшенным на 1.

Если требуется изменить содержимое открытой ячейки, новое содержимое должно предшествовать команде "←".

Пример.

@ 001000/012525 „←" (ВК, ПС)
000776/010101 "ВК" (ВК, ПС)
@

Команда "←" закрывает ячейку с адресом 1000 и открывает ячейку с адресом 776. Эта команда позволяет распечатывать содержимое массивов последовательно расположенных ячеек в сторону убывания адресов. Если эта команда используется по отношению к РСП, то она его закрывает, выполняет ВК, ПС и печатает @.

Если команда "←" используется при открытом R0, то открывается содержимое R7.

5) Команда "@" (код 100) - "открыть ячейку с абсолютным адресом". Она используется для обращения к ячейки, адресом которой является содержимое ранее открытой ячейки или РОН, после которой была введена команда "@". Эта команда выполняется при нажатии клавиши гашения на нижнем регистре.

Пример,

@ 001000/000200 „@" (ВК, ПС)
000200/000137 "ВК" (ВК, ПС)
@

Команда "@" закрывает ячейку с адресом 1000 и открывает ячейку с адресом 200.

Если команда "@" используется при открытом РСП, она закрывает его. Однако при этом открывается ячейка, адрес которой определяется содержимым регистра общего назначения, которой открывался перед РСП.

Пример.

@R3 /000200 "ВК" (ВК, ПС)
@R5 /000011 „@"
000200/000000 "ВК" (ВК, ПС)
@

3.858.382 ТО

6) Команда "–" (код 137) – "открыть ячейку с относительным адресом". Эта команда используется для открывания ячейки с адресом, определяемым как сумма трех слагаемых: содержимого уже открытой ячейки, ее адреса и +2.

Пример:

@ 001000/000200 "–" (ВК, ПС)
  001202/002525 "ВК"(ВК, ПС)
@

Команда "–" закрывает ячейку с адресом 1000 и открывает ячейку с адресом 1202. При попытке выполнить эту команду при открытом регистре общего назначения или РСП, она закрывает РОН или РСП, затем выполняет ВК, ПС и печатает знак @ .

7) Команда "R" (код 122) или "§" (044) – "регистр". Она используется для обращения к регистрам общего назначения. Если после команды "R" (или "§") и номера регистра следует команда "/", то выводится на печать содержимое указанного регистра общего назначения.

Пример:

@ R n /012345 "ВК" (ВК, ПС)
@

где R – требование обращения к регистру.
n – восьмеричный адрес (с 0 по 7).
012345 – содержимое регистра с адресом n .

Если РОН открыт, то он может быть закрыт любой выше рассмотренной командой: "ВК", "ПС", "—" или "@". Команда "—" может его закрыть, но не откроет другую ячейку с относительным адресом.

8) Команда "RS" (код 122 и код 123) – "слово состояния процессора" (Вместо команды "R" можно использовать команду "§").

Пример:

@ RS /000200 "ВК" (ВК, ПС)
@

где R – требование обращения к регистру
S – указатель обращения к РСП
000200 – содержимое РСП.

Содержимое РСП можно изменить, напечатав новое содержимое и введя команду "ВК". (Разряд Т нельзя установить пультовой командой).

9) Команда "G" (код I07) - "Пуск".

Она осуществляет пуск программы со стартового адреса, который был введен до команды "G". При этом клавиша режима работы на передней панели процессора должна находиться в положении "ПРОГРАММА".

Пример.

@ 000100 G

Счетчик команд загружается адресом I00 и начинается выполнение программы.

Перед началом выполнения программы вырабатывается канальный сигнал К СБРОС Н в течение I0 мкс.

Если клавиша режима работы находится в положении "ПУЛЬТ", то, после выработки сигнала К СБРОС Н, ЦП возвращается в режим связи с пультовым терминалом, печатает только что загруженное содержимое СК, выполняет ВК. ПС и печатает@.

I0. Команда "P" (код I20) - "продолжение". Она продолжает выполнение программы с адреса, определяемого текущим содержимым СК.

Если клавиша режима работы находится в положении "ПУЛЬТ", то выполняется одна команда программы и ЦП возвращается в режим пульта; при этом печатается содержимое СК, затем выполняется ВК, ПС и печатается@. Это используется для пошагового выполнения программы (выполнение программы с остановом после каждой команды).

II. Команда "M" (код II5) - "отладка". Эта команда используется для установления причины перехода ЦП из режима программной работы в режим связи с пультовым терминалом (в состояние "ОСТАНОВ").

Пример.

@ M C00213 (ВК, ПС)
@

ЭПМ распечатывает 6 знаков, затем выполняет ВК, ПС и печатает@. Информацию о причине останова и соответствующие им значения младшего разряда:

I) 0 - выполнение программой команды HALT (ОСТАНОВ) или перевод клавиши режима работы в положение "ПУЛЬТ";

2) I - ошибка обращения к каналу при вводе адреса вектора прерывания;

3) 2 - ошибка обращения к каналу при регенерации памяти;

4) 3 – двойная ошибка обращения к каналу (при ошибке обращения к каналу в УС – адрес несуществующей ячейки памяти);

5) 4 – ошибка обращения по несуществующему адресу микрокоманды (обращение к несуществующей странице МПЗУ).

В приведенном выше примере последняя восьмеричная цифра – 3, что означает двойную ошибку при обращении к каналу.

I2. Команда "ЗБ" (код I77) – " забой".

Она используется для исправления последнего напечатанного оператором знака. Эта команда выполняется при нажатии клавиши гашения на верхнем регистре.

Пример.

@000I00/ 077777 I23457 ЗБ 6 "ВК" (ВК, ПС)

@000I00/ I23456

Каждая команда "ЗБ" стирает только один, последний знак.

Если команда "ЗБ" выполняется последовательно 6 раз (и более) без печати нового знака, то в ячейке будет код 000000.

Пример.

@000I00/ 07777 I23457 ЗБЗБЗБЗБЗБЗБ ВК ВК, ПС

@000I00/ 000000

I3. Команда "L" (код II4) – "загрузка".

Она используется для ввода программы абсолютного загрузчика. Перед тем, как подать команду "L", необходимо ввести в ПГ адрес регистра состояния считывающего устройства.

Например:

@ I77550L

где I77550 – адрес регистра состояния устройства считывания с перфоленты.

По окончании ввода ленты абсолютного загрузчика ЭВМ выводит на печать стартовый адрес программы абсолютного загрузчика, если клавиша режима работы на передней панели процессора находится в положении "ПРОГРАММА":

@ I77550L

037500

где 037500 – стартовый адрес абсолютного загрузчика для 8К памяти.

Если клавиша режима работы находилась в положении "ПУЛЬТ", по окончании ввода ленты абсолютного загрузчика напечатается 037724, и для получения стартового адреса абсолютного загруз-

чика необходимо перевести клавишу режима работы в положение "ПРОГРАММА" и подать команду "Р".

Область памяти, куда вводится абсолютный загрузчик и, следовательно, его стартовый адрес зависит от объема памяти. Для объема памяти 4К стартовый адрес 017500 ; 8К - 037500; 12К - 057500; 16К - 077500; 20К - 117500; 24К - 137500; 28К - 157500.

После подачи команды "L" ЦП проверяет объем памяти и вводит абсолютный загрузчик в последний банк памяти. ВВОД и программы с помощью команды "L" будет происходить при любом положении клавиши режима работы. При ошибке обращения к каналу ввод прекращается, печатается ?, выполняется ВК, ПС и печатается @. Если перед командой "L" адрес устройства не напечатан, ЦП обратится к устройству, адрес регистра состояния которого - 000000.

14. Команда "СКП S" (код 023) используется для отладочных целей и не предназначена для пользователя. Здесь приводится описание этой команды с той целью, чтобы пользователь понял реакцию ЦП при случайном ее исполнении. Если эта команда подана, ЦП ожидает ввода еще двух символов, коды которых образуют двухбайтовый двоичный адрес. Символ, введенный первым, является его старшим байтом, а символ, введенный вторым - младшим. Начиная с этого адреса, ЦП выводит на печать содержимое пяти ячеек памяти (десять байтов). При случайном вводе этой команды рекомендуется напечатать два символа (например, СКП и #), чтобы завершить ее выполнение. Тогда ЭПМ выполнит ВК, ПС и напечатает @.

4.7. Организация обмена информацией с внешними устройствами.

Обмен информацией между центральным процессором и внешними устройствами выполняется при помощи стандартных циклов обращения к каналу (см.п.4.I.I.5). Для организации обмена каждое внешнее устройство должно иметь один или несколько регистров (регистры данных, регистры состояний и т. д.), адреса которых определяет пользователь. Эти регистры могут быть построены на триггерах и выполнять функции хранения, или же состоять из простых вентилей, открываемых в момент прохождения через них информации.

Как правило, регистры ВУ имеют четные адреса, однако при помощи байтовых команд можно обращаться к любому байту I6-разрядного регистра.

Восемь I6-разрядных регистров с адресами I77550-I77566 зарезервированы для регистров стандартных устройств ввода-вывода информации и пользователю не рекомендуется использовать их для других целей.

Различные разряды регистров ВУ могут выполнять различные функции. Некоторые из них могут использоваться как для записи, так и для считывания информации, другие - только для записи или только для считывания. Типичным примером разряда, используемого для считывания и записи, является разряд разрешения прерывания регистра состояния ВУ. Примером разряда, только принимающего информацию, является разряд пуска, а разрядом, используемым только для считывания, разряд ошибки регистра состояния ВУ.

4.7.I. Форматы регистров ВУ.

Регистры данных ВУ, как правило, являются обычными накопительными регистрами и их формат определяется только требованиями пользователя.

Рекомендуемый формат регистров состояния ВУ показан на рис.39. Данный формат не является обязательным. Он приводится только для облегчения разработки и организации работы различных устройств в системе с единым каналом передачи информации, а также для наиболее эффективного использования системы команд.

Заметим, что большинство регистров состояния ВУ имеют меньшее, чем I6 количество разрядов.

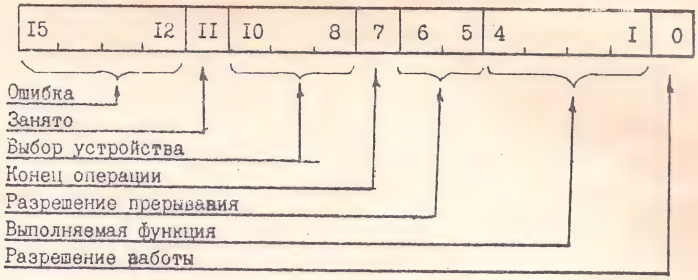


Рис. 39. Формат регистра состояния ВУ.

Ошибка (I5-I2) - Эти разряды используются для сигнализации о наличии ошибки (разряд I5) и характере ошибки (разряды I4-I2). Они устанавливаются внешним устройством и должны очищаться ЦП для того, чтобы разрешить выполнение операций. Если для указания характера ошибки недостаточно трех разрядов, они могут быть использованы для указания номеров специальных регистров ошибок. Большинство устройств используют разряд I5 для вызова прерывания программы по ошибке (если установлен разряд разрешения прерывания).

Занято (II) - устанавливается для сигнализации о том, что внешнее устройство занято выполнением внутренних операций и не может принять участия в обмене данными. Для некоторых устройств этот разряд не является необходимым, его функции выполняет разряд "Конец операции".

Выбор устройства (I0-08) - эти разряды используются для выбора различных устройств, подключенных к общему устройству управления. Такими устройствами могут быть накопители на магнитных дисках, подключенные к общему контролеру.

Конец операции (07) - устанавливается внешним устройством, когда оно готово для обмена данными. Обычно используется для требования прерывания.

Разрешение прерывания (06,05) - эти разряды устанавливаются центральным процессором для разрешения прерывания и передачи адреса вектора прерывания внешним устройством. Разряд 05 используется в том случае, если регистр состояния имеет два разряда "Конец операции".

Выполняемая функция (04-01) - эти разряды определяют операцию, выполняемую ВУ (считывание, запись, перфорация, поиск зоны и т. д.).

Разрешение работы (00) - этот разряд используется для запуска внешнего устройства.

4.7.2. Назначение регистров ВУ.

Каждое внешнее устройство может иметь несколько различных регистров. В этом разделе рассматривается назначение некоторых основных регистров внешних устройств и порядок, в котором им присваиваются адреса.

Регистр состояния (РС) содержит информацию об операции, выполняемой ВУ, характеризует состояние ВУ и участвует в операциях по предоставлению прерывания.

Регистр данных (РД) используется при обмене данными между ЦП и ВУ.

Регистр адреса памяти (РАП) используется для хранения адресов ячеек памяти при передаче массивов слов или байтов. Содержимое этого регистра наращивается после передачи каждого слова.

Регистр счета слов (РСС) используется для управления передаваемого массива данных. Информация в РСС заносится из ЦП.

Регистр адреса устройства (РАУ) используется для хранения номера дорожки или блока в запоминающих устройствах большой емкости.

Если в устройстве имеется несколько регистров, им рекомендуется присваивать последовательные адреса в следующем порядке:

РС I
РС 2
РД I
РД 2
РД 3
РД 4
РАП
РСС
РАУ

4.8. Подключение устройств пользователя к ЭВМ.

Для подключения устройств пользователя к ЭВМ можно использовать стандартное устройство параллельного обмена. Однако если характеристики устройства параллельного обмена не удовлетворяют пользователя, он может разработать свое устройство связи с каналом ЭВМ. Рекомендации по разработке таких устроств приводятся в данном подразделе.

4.8.I. Приемо-передатчики канала ЭВМ.

Для связи с каналом ЭВМ пользователь должен применять специальные ИС приемников и передатчиков, аналогичные тем, которые используются в центральном процессоре. При этом следует помнить, что линии канала КДА (00 - I5) Н выполнены с использованием приемо-передатчиков с открытым коллектором.

Устройства пользователя, подключаемые к каналу ЭВМ, могут работать в программном режиме, в режиме прерывания программы и в режиме прямого доступа к памяти (ПДП).

4.8.2.Программный режим работы

Логика устройства пользователя,работающего в программном режиме,показана на рис.40.Так как количество и функциональное назначение регистров,а также логика,связанная непосредственно с периферийным устройством,определяются пользователем, на рис.40 показана только логика управления,связанная с каналом ЭВМ.

В ЭВМ последние 4К адресов зарезервированы для внешних устройств,поэтому при обращении к ВУ центральный процессор будет вырабатывать сигнал К ВУ Н.Сигналы К ВУ Н и К ДА(01-12) Н будут обеспечивать выбор любого 16-разрядного регистра ВУ,адрес которого находится в пределах 160000-177776.Адреса регистров ВУ задаются перемычками или переключателями на входах схемы сравнения адреса,и их конфигурация определяется пользователем.

На рис.40 показано устройство управления,которое имеет 4 программно доступных регистра.Разряды адреса 03-12 определяют выбор устройства,разряды 01,02 выбирают регистр,а разряд 00 определяет,к какому байту регистра идет адресация.Сигнал К СИА Н информирует о том,что на линиях К ДА(00-15) установлен адрес,и может быть использован для запоминания сигнала ВЫБОР и разрядов адреса 00-02.

Если сигнал ВЫБОР после окончания адресной части цикла канала будет активным,то после поступления сигналов К ВВОД Н или К ВЫВОД Н,а также сигнала К БАЙТ Н,дешифратор вырабатывает сигналы выбора регистров.Эти сигналы могут быть использованы для стробирования регистров,причем каждый сигнал определяет характер обращения к ним.

Если устройство управления имеет более четырех программно доступных регистров,то выбор регистра будет осуществляться при помощи необходимого количества младших разрядов адреса.Вместо дешифратора пользователь может применять ИС ПЗУ или ППЗУ.

После того,как данные будут приняты или переданы устройством пользователя,логика управления должна выработать сигнал К СИП Н.Если этот сигнал не будет передан в течение приблизительно 10 мкс,ЦП переходит на подпрограмму обслуживания внутреннего прерывания с адресом вектора 4.Если устройство пользователя способно работать в режиме прерывания программы,то сигнал К СИП Н должен вырабатываться и при передаче адреса вектора

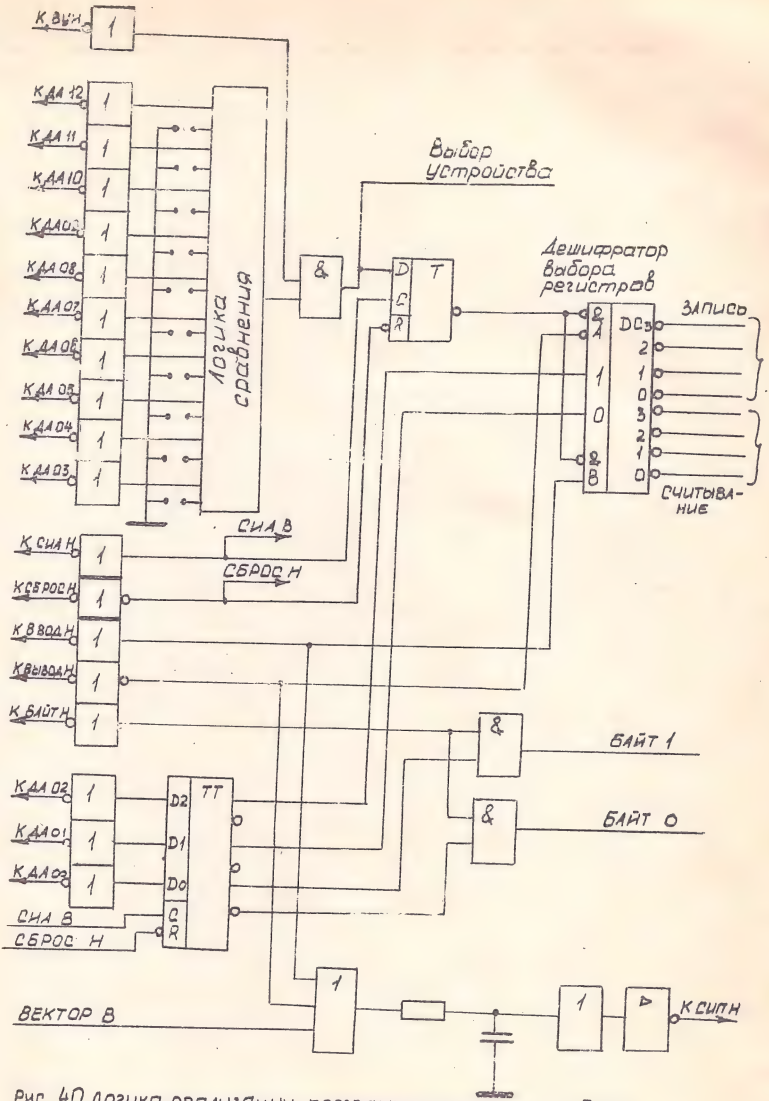


Рис. 40 Логика реализации программного режима работы.

3.858.382 ТО

прерывания.

4.8.3.Режим прерывания программы

Устройство пользователя,способное работать в режиме прерывания программы,должно обеспечивать выработку сигнала требования прерывания(К ТПР Н), прием и передачу сигнала предоставления прерывания (К ПРР Н),формирование и передачу адреса вектора прерывания.

Основные логические функции,необходимые для обеспечения работы устройства пользователя в режиме прерывания,выполняет схема,которая приведена на рис.4I.Она позволяет вызывать прерывание программы по двум независимым запросам.

Логика прерывания содержит четыре триггера РПР А,РПР Б, ТПР А,ТПР Б,которые соответствуют разрядам одного или двух регистров состояния устройства пользователя.Управление этими триггерами осуществляется при помощи сигналов выбора регистров, которые были описаны в п.4.8.2.

Когда устройство пользователя готово для обслуживания в режиме прерывания программы,оно вырабатывает активный уровень сигнала ГОТОВО А В.Этот сигнал,объединяясь по "И" с РПР А(I) В, устанавливает триггер ТПР А,разрешая тем самым выработку сигнала К ТПР Н.Однако,если сигнал ГОТОВО А будет сброшен,это вызовет очистку линии К ТПР Н.Центральный процессор,получив сигнал К ТПР Н,вырабатывает сигналы К ВВОД Н и К ПРР Н (см. п. 4.I.I.5).Устройство пользователя принимает эти сигналы, очищает триггер ПРР А и запрещает дальнейшее распространение сигнала К ПРР Н по каналу ЭВМ.Кроме того,триггер ПРР А вырабатывает активные сигналы ВЕКТОР Н и ВЕКТОР А В,первый из которых разрешает передачу адреса вектора прерывания,а второй участвует в его формировании.Сигнал ВЕКТОР А В обычно поступает на мультиплексор выходных данных и определяет адрес вектора А или Б.

В случае одновременного возникновения требований прерывания А и Б,первым будет удовлетворяться требование А.Таким образом,устройство,вырабатывающее требование А,имеет более высокий приоритет обслуживания.

4.8.4. Режим прямого доступа к памяти

Устройство пользователя,способное работать в режиме ПДП, должно обеспечивать формирование адресов ячеек памяти,прием и передачу данных,контролировать длину передаваемого массива

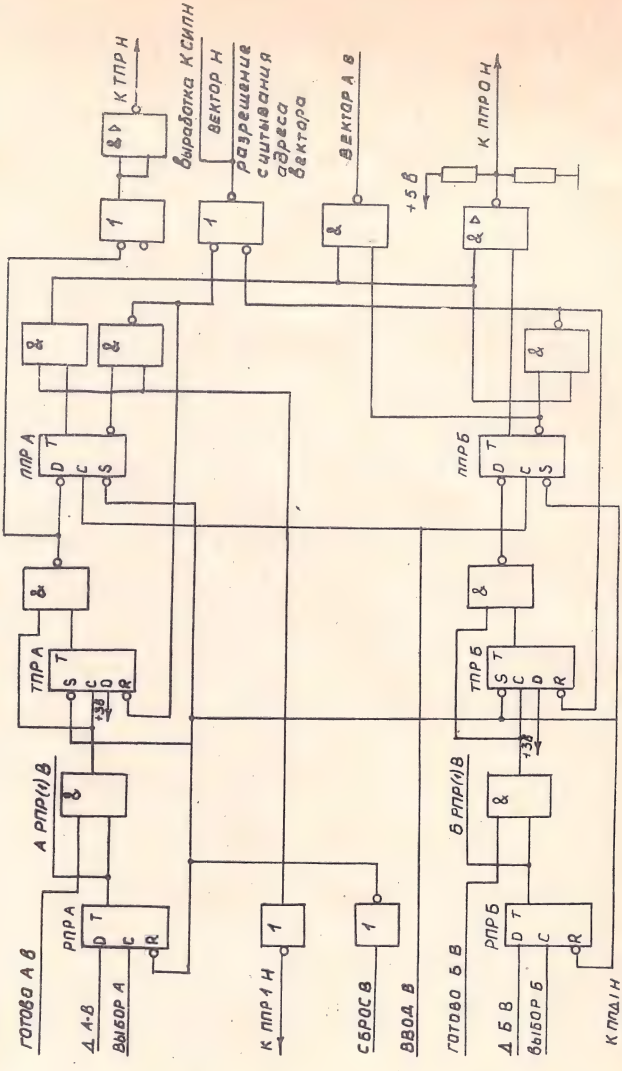


Рис. 41 Логика управления прерыванием программы.

3.858.382 ТО

данных. Эти функции выполняются соответствующими регистрами устройства пользователя. Управление всеми регистрами может быть осуществлено при помощи сигналов выбора регистров (см. п. 4.8.2). Кроме того, устройство ПДП должно иметь логику предоставления ПДП, показанную на рис. 42.

Если устройству пользователя требуется прямой доступ к памяти, оно вырабатывает ТРБ В, который должен оставаться активным до тех пор, пока не закончится текущий цикл обращения к каналу. В это время сигнал ТРБ В будет определять выработку сигнала К ТПР Н. После того, как предыдущее активное устройство освободит канал, ЦП выработает сигнал К ППД Н. Сигнал К ППД Н установит триггеры подтверждения выбора (ПВ) и прямого доступа к памяти (ППД). Установленный триггер ППД прекратит выработку сигнала К ТПД Н. Он также может быть использован для разрешения устройству пользователя начать свой цикл обращения к каналу. Триггер ПВ будет определять выработку сигнала ПВ Н, который должен оставаться активным до тех пор, пока устройство пользователя не закончит обмен данными в режиме ПДП.

Если устройство пользователя не требовало прямого доступа к памяти, триггер ПВ будет очищен, а сигнал ППД Н будет передан к следующему устройству ПДП.

Если в системе используется динамическая память, для которой необходима регенерация, устройство ПДП должно использовать одиночные циклы обращения к каналу. При необходимости обмена массивами данных осуществлять регенерацию памяти должно устройство ПДП.

При разработке устройств, подключаемых к каналу ЭВМ, следует помнить, что линия К ПРР и К ППД проходят последовательно через все устройства. Если устройство пользователя не предназначено для работы в режимах прерывания или прямого доступа, то должны быть предусмотрены специальные перемычки, которые позволят следующим устройствам принять сигналы К ПРР Н и К ППД Н.

4.9. Дополнительные устройства к ЭВМ

Как уже отмечалось ранее, ЭВМ представляет собой систему модулей, объединенных каналом ЭВМ. Поэтому пользователю представляется возможность самому определить необходимую конфигурацию системы. Например, объем оперативного ЗУ может быть увеличен до 28К 16-разрядных слов путем простого добавления необходимого числа полупроводниковых запоминающих устройств. При этом не требуется никаких специальных устройств сопряжения.

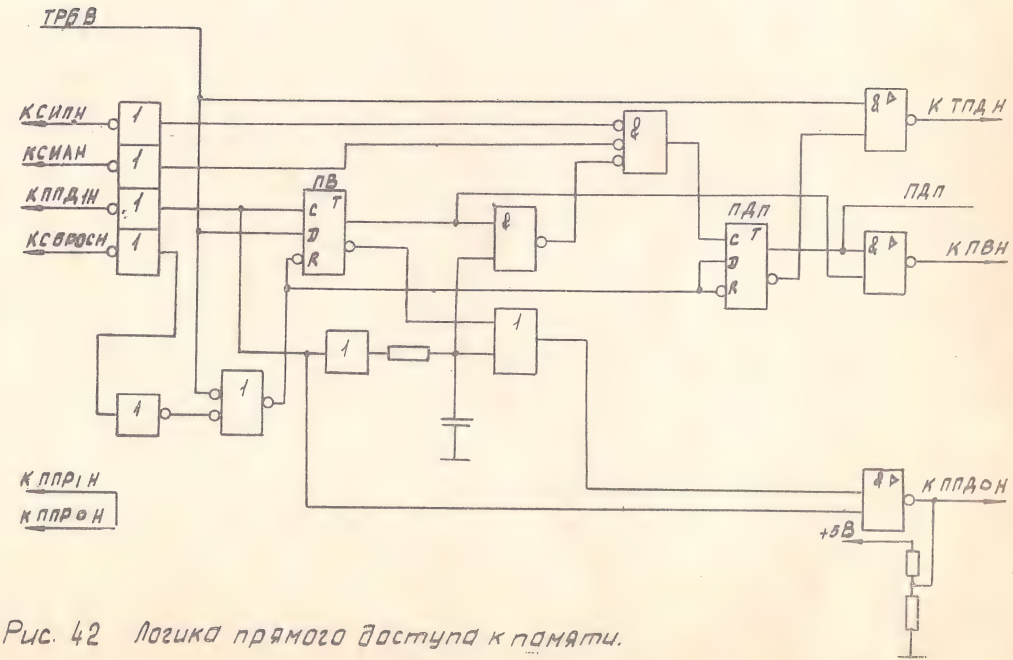


Рис. 42 Логика прямого доступа к памяти.

Кроме того, пользователь может применить устройство параллельного обмена в количестве, определяемом конкретным использованием.

На рис.43 показана блок-схема ЭВМ максимального состава. Следует заметить, что адресное пространство оперативного запоминающего устройства может быть использовано перепрограммируемыми постоянными ЗУ или обычным ПЗУ. Количество устройств параллельного обмена и устройств управления пользователя определяется требованиями пользователя. Стандартный таймер, вырабатывающий периодический сигнал с частотой 50 ± I Гц, имеется в блоке питания. Однако пользователь может сам разработать таймер с необходимыми ему характеристиками и подключить к каналу ЭВМ.

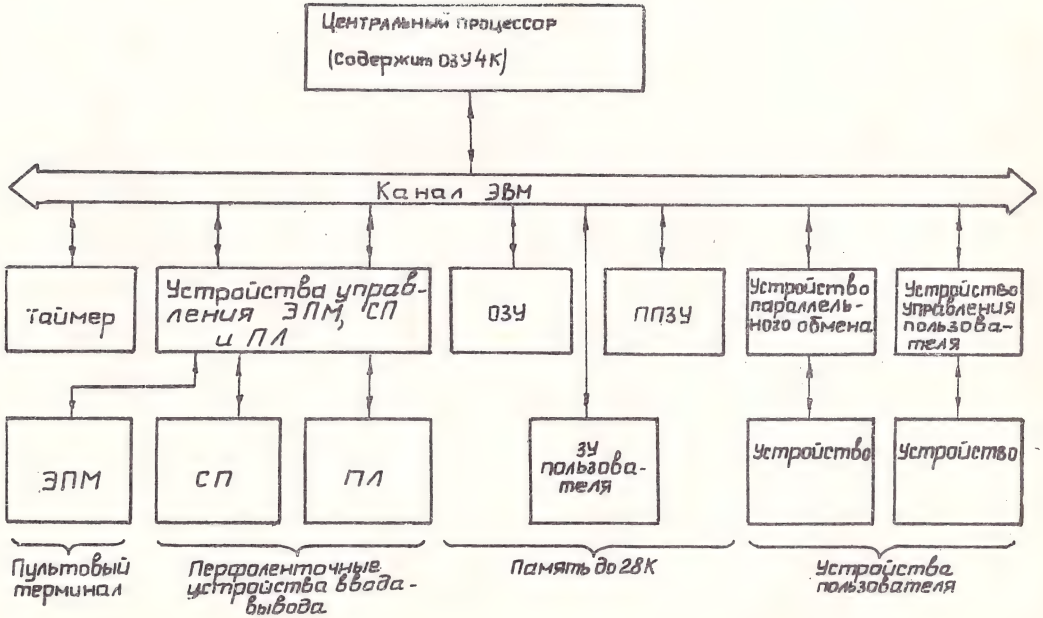


Рис. 43 Блок-схема ЭВМ „Электроника 60М"

5. МАРКИРОВАНИЕ.

5.I. Центральный процессор имеет маркировку, содержащую:
- товарный знак предприятия-изготовителя или товарный знак внешнеторговой организации;
- сокращенное обозначение;
- заводской номер;
- месяц и год выпуска.

6.. УКАЗАНИЕ МЕР БЕЗОПАСНОСТИ.

6.I. К работе с центральным процессором допускаются лица, изучившие настоящую инструкцию, инструкцию по технике безопасности при работе на данном оборудовании, а также прошедшие местный инструктаж по безопасности труда.

6.2. ЦП может обслуживать один инженер-оператор, имеющий квалификационную группу не ниже III.

6.3. Съем, установку, а также ремонт ЦП производить только при отключенном питании.

7. ПОРЯДОК УСТАНОВКИ.

7.1. Нормальные условия эксплуатации ЦП в составе ЭВМ следующие:

а) температура окружающего воздуха, ^{o}C 20 ± 5;
б) относительная влажность окружающего воздуха,% 65 ± I5;
в) атмосферное давление, кПа от 84,0 до I06,7
(мм рт. ст.) (от 630 до 800).

7.2. Запрещается эксплуатировать ЦП в помещениях с химически агрессивной средой.

7.3. ЦП после транспортирования в зимнее время года выдержите в упаковке в течение 24 часов в помещении, где оно будет эксплуатироваться, затем распакуйте.

7.4. Произведите внешний осмотр ЦП, убедитесь в отсутствии механических повреждений печатных проводников и элементов устройства.

7.5. Установите с помощью перемычек необходимый режим работы ЦП, руководствуясь табл. I7.

7.6. Установите ЦП в гнездо монтажной панели ЭВМ с помощью рычагов не допуская перекосов.

7.7. Произведите проверку работоспособности ЦП согласно

8. ПОРЯДОК РАБОТЫ.

8.I. Обслуживание ЦП осуществляется тем же персоналом, который эксплуатирует ЭВМ.

8.2. Для выявления неисправностей в работе ЦП необходимо иметь;

- амперволтметр Ц4341;
- осциллограф СI-64.

8.3. При эксплуатации ЦП не следует касаться руками без статического браслета выводов БИС.

9. ПРОВЕРКА ТЕХНИЧЕСКОГО СОСТОЯНИЯ.

9.I. Проверка технического состояния ЦП осуществляется согласно инструкции по эксплуатации на ЭВМ.

I0. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ.

I0.I. Техническое обслуживание ЦП осуществляется согласно инструкции по эксплуатации на ЭВМ.

Приложение

Список команд ЭВМ "Электроника 60"

Таблица 19

Мнемоника	Команда	Код	Лист
I	2	3	4
	Одноадресные команды		
CLR(B)	Очистка	# 050D D	83
COM(B)	Инвертирование	# 051D D	84
INC(B)	Прибавление единицы	# 052D D	85
DEC(B)	Вычитание единицы	# 053D D	86
NEG(B)	Изменение знака	# 054D D	87
TST(B)	Проверка	# 057D D	88
ASR(B)	Арифметический сдвиг вправо	# 062D D	89
ASL(B)	Арифметический сдвиг влево	# 063D D	90
ROR(B)	Циклический сдвиг вправо	# 060D D	92
ROL(B)	Циклический сдвиг влево	# 061D D	93
ADC (B)	Прибавление переноса	# 055D D	95
SBC(B)	Вычитание переноса	# 056 D D	96
SXT	Расширение знака	0067D D	96
SWAB	Перестановка байтов	0003D D	97
MFPS	Чтение ССП	I067D D	98
MTPS	Запись ССП	I064S S	99
	Двухадресные команды		
MOV(B)	Пересылка	# 1SSD D	I00
CMP(B)	Сравнение	# 2SSD D	I0I
ADD	Сложение	06SSD D	I02
SUB	Вычитание	I6SSD D	I03
BIT(B)	Проверка разрядов	# 3SSD D	I04
BIC(B)	Очистка разрядов	# 4SSD D	I05
BIS(B)	Логическое сложение	# 5SSD D	I06
XOR	Исключающее ИЛИ	074RD D	I07

Мнемоника	Команда	Код	Лист
I	2	3	4
	Команды управления программой		
BR	Ветвление безусловное	000400	I09
BNE	Ветвление, если не равно (нулю)	001000	II0
BEQ	Ветвление, если равно (нулю)	001400	II0
BPL	Ветвление, если плюс	I00000	III
BMI	Ветвление, если минус	I00400	II2
BVC	Ветвление, если нет арифметического переполнения	I02000	II2
BVS	Ветвление, если арифметическое переполнение	I02400	II2
BCC	Ветвление, если нет переноса	I03000	II2
BCS	Ветвление, если перенос	I03400	II3
BGE	Ветвление, если больше и равно (нулю)	002000	II4
BLT	Ветвление, если меньше (нуля)	002400	II5
BGT	Ветвление, если больше (нуля)	003000	II5
BLE	Ветвление, если меньше или равно (нулю)	003400	II6
BHI	Ветвление, если больше	I0I000	II6
BLOS	Ветвление, если меньше или равно	101400	II7
BHIS	Ветвление, если больше или равно	I03000	II7
BLO	Ветвление, если меньше	I03400	II7
JMP	Безусловный переход	0001D D	II8
JSP	Обращение к подпрограмме	004RD D	II9
RTS	Возврат из подпрограммы	00020R	I2I
MARK	Восстановление УС	0064NN	I22

1	2	3	4
SOB	Вычитание единицы и ветвление	077RNNN	I23
	Команды прерывания программы		
EMT	Командное прерывание для системных программ	I04000--I04377	I24
TRAP	Командное прерывание	I04400--I04777	I25
IOT	Командное прерывание для ввода-вывода	000004	I25
BPT	Командное прерывание для отладки	000003	I26
RTI	Возврат из прерывания	000002	I27
RTT	Возврат из прерывания	000006	I27
	Команды управления машиной		
HALT	Останов	000000	I27
WAIT	Ожидание	00000I	I28
RESET	Сброс внешних устройств	000005	I29
	Команды изменения признаков		
CLN	Очистка N	000250	I29
CLZ	Очистка Z	000244	I29
CLV	Очистка V	000242	I29
CLC	Очистка C	00024I	I30
CCC	Очистка всех разрядов (N,Z,V,C)	000257	I30
SEN	Установка N	000270	I30
SEZ	Установка Z	000264	I30
SEV	Установка V	000262	I30
SEC	Установка C	00026I	I30
SCC	Установка всех разрядов (N,Z,V,C)	000277	I30

I	2	3	4
NOP	Нет операции	000240	I30
	Команды расширенной арифметики и плавающей запятой		
MUL	Умножение	070RSS	I39
DIV	Деление	07IRSS	I40
ASH	Арифметический сдвиг	072RSS	I4I
ASHC	Арифметический сдвиг двойного слова	073RSS	I43
FADD	Сложение с плавающей запятой	07500R	I49
FSUB	Вычитание с плавающей запятой	0750IR	I49
FMUL	Умножение с плавающей запятой	07502R	I50
FDIV	Деление с плавающей запятой	07503R	I5I

ПРИЛОЖЕНИЕ